Chapter 4: Manipulating Routing Updates

CCNP ROUTE: Implementing IP Routing

Cisco | Networking Academy®
Mind Wide Open™

Chapter 4 Objectives

- Describe network performance issues and ways to control routing updates and traffic.
- Describe the purpose of and considerations for using multiple routing protocols in a network.
- Configure and verify route redistribution of multiple protocols.
- Describe, configure and verify various methods for controlling routing update traffic.

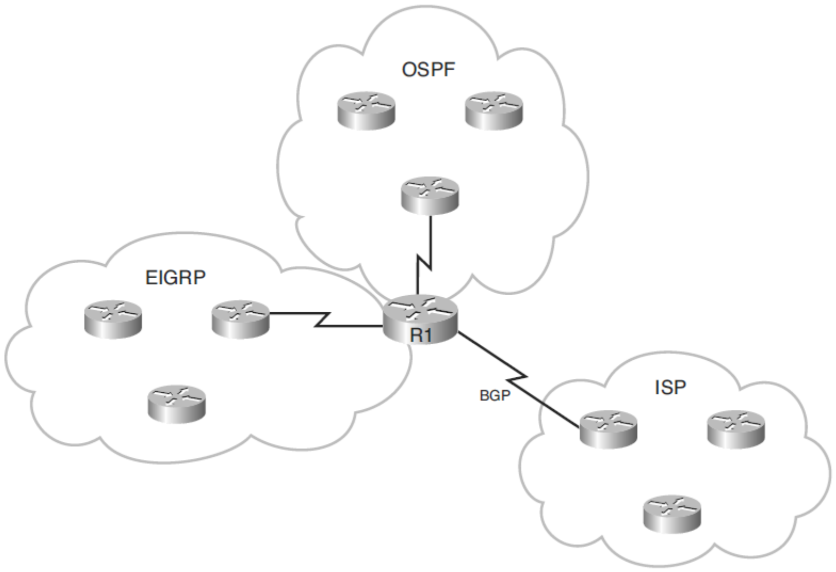
Assessing Network Routing Performance Issues

Common Routing Performance Issues

- Excessive routing updates
  - CPU utilization can easily spike during this processing depending on:
    - The size of the routing update
    - The frequency of the updates
    - The Layer 3 network design
- The presence of any incorrectly configured route maps or filters
- The number of routing protocols running in the same autonomous system

Running Multiple Protocols

- Different routing protocols were not designed to interoperate with one another.
  - Each protocol collects different types of information and reacts to topology changes in its own way.
- Running muliple routing protocols increases CPU utilization and requires more memory resources to maintain all the topology, database and routing tables.



Routing Protocol Performance Solutions

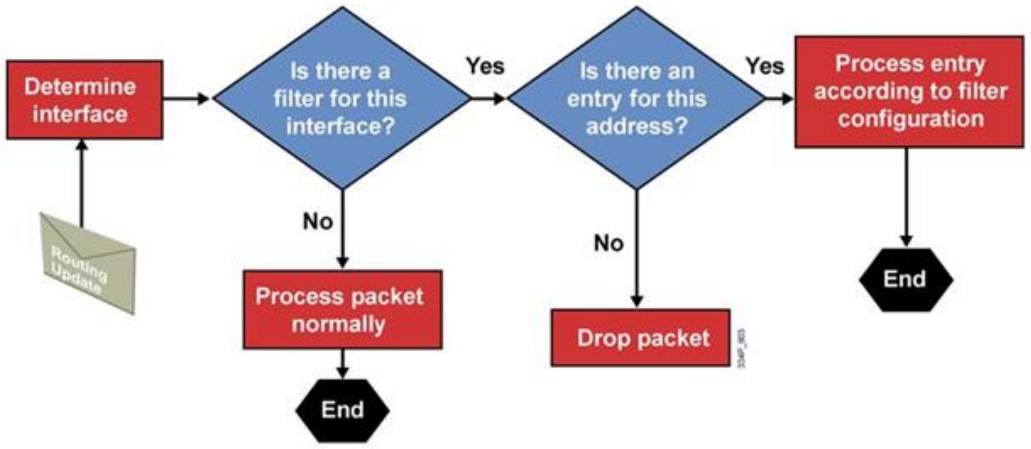
- Design changes, such as limiting the number of routing protocols used.
- Using passive interfaces to prevent routing protocol updates from being advertised out an interface.
- Route filtering techniques to block specific routes from being advertised:
  - Access control lists (ACLs)
  - Route maps
  - Distribute lists
  - Prefix lists

Route Filtering

- Using route maps, distribute lists, or prefix lists instead of access lists provides greater route filtering flexibility.
- Filters can be configured to:
  - Prevent updates through router interfaces.
  - Control the advertising of routes in routing updates.
  - Control the processing of routing updates.
- If filters are not configured correctly or if filters are applied to wrong interfaces, network performance issues may occur.

Route Filtering Process

1. A router stores the incoming routing update in the buffer and triggers a decision.
2. Is there an incoming filter applied to this interface?
   - If no, then the routing update packet is processed normally.
3. Otherwise, is there an entry in the filter matching the routing update packet?
   - If no, then the routing update packet is dropped.
4. Otherwise, the router processes the routing update according to the filter.


Determine interface
Routing Update
Is there a filter for this interface?
Yes
No
Process packet normally
End
Is there an entry for this address?
Yes
No
Drop packet
Process entry according to filter configuration
End

Using Multiple Routing Protocols on a Network

Simple to Complex Networks

- Simple routing protocols work well for simple networks.
  - Typically only require one routing protocol.
- Running a single routing protocol throughout your entire IP internetwork is desirable.
- However, as networks grow they become more complex and large internetworks may have to support several routing protocols.
  - Proper inter-routing protocol exchange is vital.

Why have multiple routing protocols?

- Interim during conversion
  - Migrating from an older IGP to a new IGP.
- Application-specific protocols
  - One size does not always fit all.
- Political boundaries
  - Multiple departments managed by different network administrators
  - Groups that do not work well with others
- Mismatch between devices
  - Multivendor interoperability
  - Host-based routers
- Company mergers

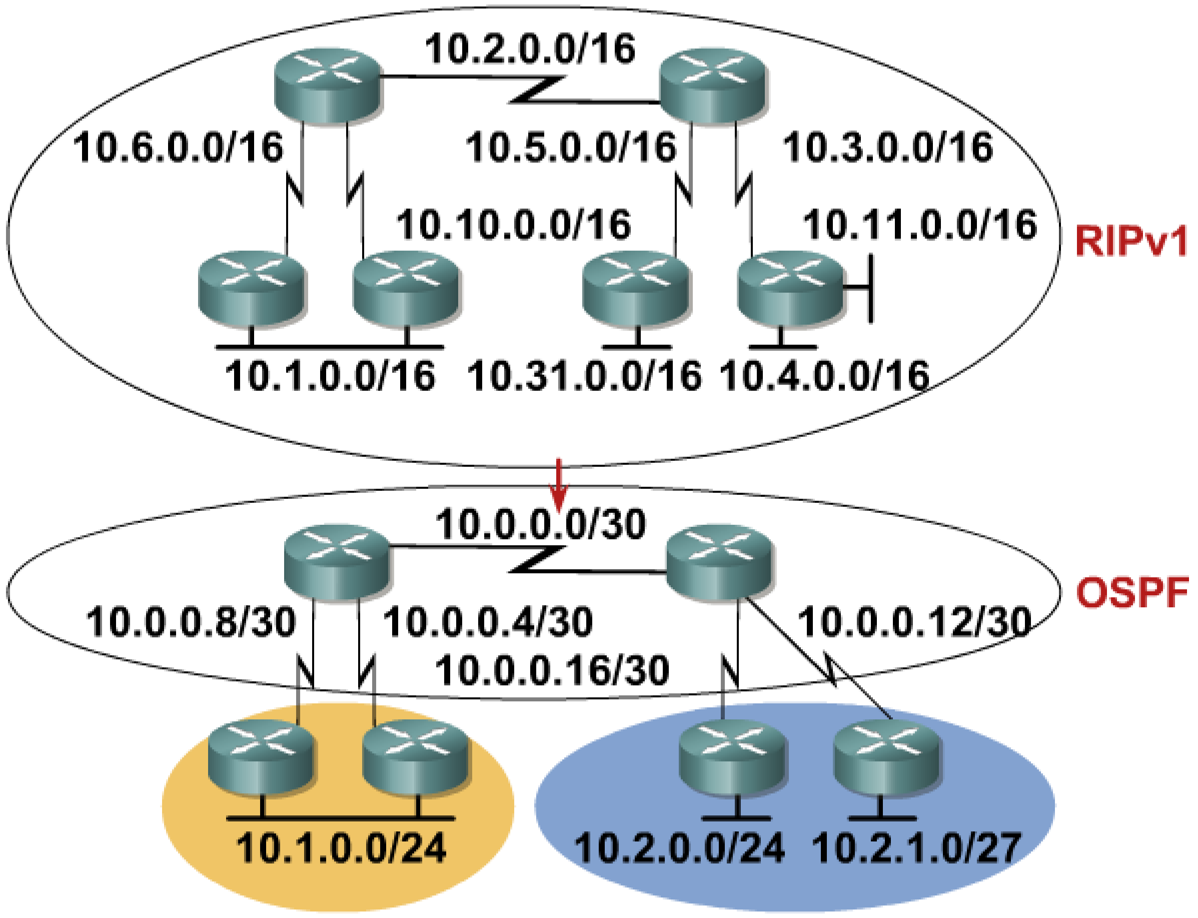
Complex Networks

- Complex networks require careful routing protocol design and traffic optimization solutions, including the following:
  - Redistribution between routing protocols
  - Route filtering (covered in the next chapter)
  - Summarization (covered in EIGRP and OSPF)

Redistribution

- Cisco routers allow different routing protocols to exchange routing information through a feature called *route redistribution.*
  - Route redistribution is defined as the capability of boundary routers connecting different routing domains to exchange and advertise routing information between those routing domains (autonomous systems).

Route Redistribution Example



- FLSM to VLSM
- Hierarchical Addressing
- Hierarchical Areas

Redistributed Routes

- Redistribution is always performed outbound; the router doing redistribution does not change its routing table.
- The boundary router’s neighbors see the redistributed routes as external routes.
- Routes must be in the routing table for them to be redistributed.

Redistribution Considerations

- The key issues that arise when using redistribution:
  - **Routing feedback (loops)**
    - If more than one boundary router is performing route redistribution, then the routers might send routing information received from one autonomous system back into that same autonomous system.
  - **Incompatible routing information**
    - Each routing protocol uses different metrics to determine the best path therefore path selection using the redistributed route information might not be optimal.
  - **Inconsistent convergence times**
    - Different routing protocols converge at different rates.
- Good planning should solve the majority of issues but additional configuration might be required.
  - Some issues might be solved by changing the administrative distance, manipulating the metrics, and filtering using route maps, distribute lists, and prefix lists.

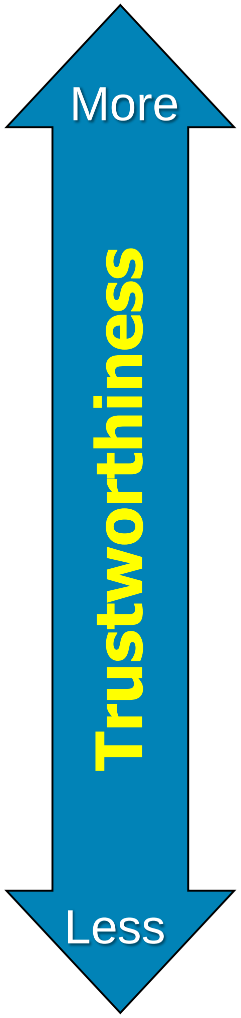
Selecting the Best Route

Routers use the following two parameters to select the best path:

- **Administrative distance**:
  - Used to rate a routing protocol’s believability (also called its trustworthiness).
  - This criterion is the first thing a router uses to determine which routing protocol to believe if more than one protocol provides route information for the same destination.
- **Routing metric:**
  - The routing metric is a value representing the path between the local router and the destination network, according to the routing protocol being used.
  - The metric is used to determine the routing protocol’s “best” path to the destination.

Cisco IOS Administrative Distance

Routing Protocol	Default Administrative Distance Value
Connected interface	0
Static route out an interface	1
Static route to a next-hop address	1
EIGRP summary route	5
External BGP	20
Internal EIGRP	90
IGRP	100
OSPF	110
IS-IS	115
RIPv1 and RIP v2	120
Exterior Gateway Protocol (EGP)	140
On-Demand Routing (ODR)	160
External EIGRP	170
Internal BGP	200
Unknown	255



Routing Metric

- A boundary router must be capable of translating the metric of the received route into the receiving routing protocol.
  - Redistributed route must have a metric appropriate for the receiving protocol.
- The Cisco IOS assigns the following default metrics when a protocol is redistributed into the specified routing protocol:

Protocol That Route Is Redistributed Into …	Default Seed Metric
RIP	**0** (interpreted as infinity)
IGRP / EIGRP	**0** (interpreted as infinity)
OSPF	**20** for all except BGP routes (BGP routes have a default seed metric of 1)
IS-IS	**0**
BGP	BGP metric is set to IGP metric value

Defining a Seed Metric

- A seed metric, different than the default metric, can be defined during the redistribution configuration.
  - After the seed metric for a redistributed route is established, the metric increments normally within the autonomous system.
    - The exception to this rule is OSPF E2 routes.
- Seed metrics can be defined in two ways:
  - The **`default-metric`** router configuration command establishes the seed metric for all redistributed routes.
  - The **`redistribute`** can also be used to define the seed metric for a specific protocol.

OSPF Seed Metric Example #1

```
R3(config)# router rip
R3(config-router)# network 172.18.0.0
R3(config-router)# network 172.19.0.0
R3(config-router)# router ospf 1
R3(config-router)# network 192.168.2.0 0.0.0.255 area 0
R3(config-router)# redistribute rip subnets metric 30
R3(config-router)#
```

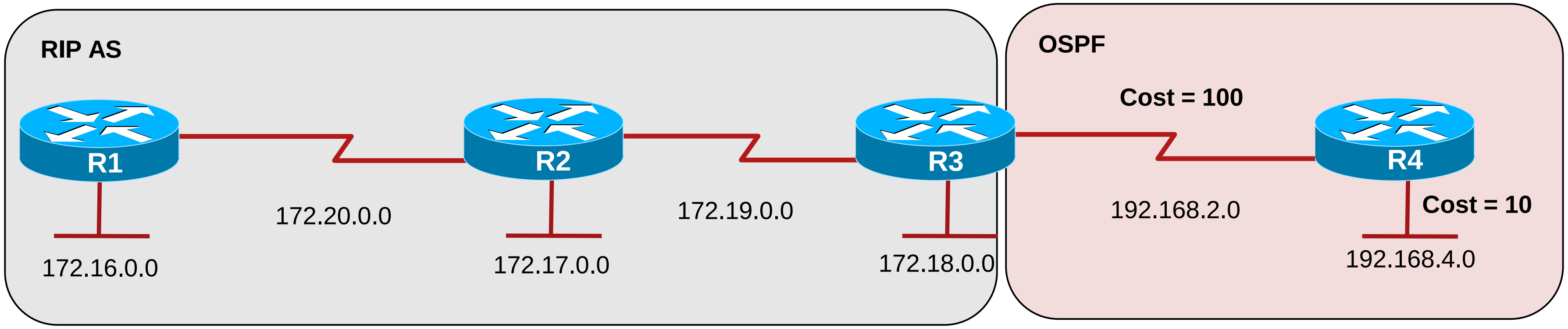


Table R1

```
C 172.16.0.0
C 172.20.0.0
R [120/1] 172.17.0.0
R [120/1] 172.19.0.0
R [120/2] 172.18.0.0
```

Table R2

```
C 172.17.0.0
C 172.19.0.0
C 172.20.0.0
R [120/1] 172.16.0.0
R [120/1] 172.18.0.0
```

Table R3

```
C 172.18.0.0
C 172.19.0.0
R [120/1] 172.17.0.0
R [120/1] 172.20.0.0
R [120/2] 172.16.0.0
C 192.168.2.0
O [110/110] 192.168.1.0
```

Table R4

```
C 192.168.1.0
C 192.168.2.0
O E2 [110/30] 172.16.0.0
O E2 [110/30] 172.17.0.0
O E2 [110/30] 172.18.0.0
O E2 [110/30] 172.19.0.0
O E2 [110/30] 172.20.0.0
```

OSPF Seed Metric Example #2

```
R3(config)# router rip
R3(config-router)# network 172.18.0.0
R3(config-router)# network 172.19.0.0
R3(config-router)# router ospf 1
R3(config-router)# network 192.168.2.0 0.0.0.255 area 0
R3(config-router)# redistribute rip subnets
R3(config-router)# default-metric 30
```

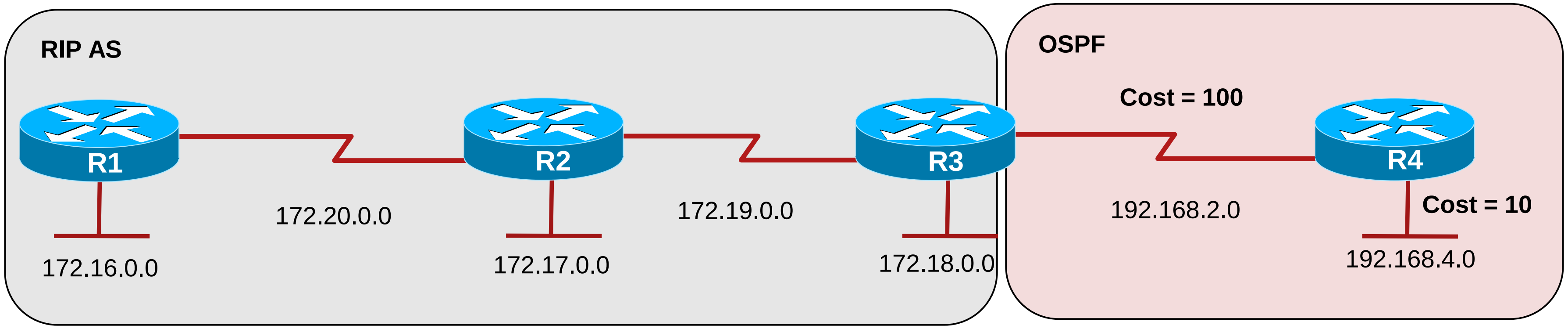


Table R1

```
C 172.16.0.0
C 172.20.0.0
R [120/1] 172.17.0.0
R [120/1] 172.19.0.0
R [120/2] 172.18.0.0
```

Table R2

```
C 172.17.0.0
C 172.19.0.0
C 172.20.0.0
R [120/1] 172.16.0.0
R [120/1] 172.18.0.0
```

Table R3

```
C 172.18.0.0
C 172.19.0.0
R [120/1] 172.17.0.0
R [120/1] 172.20.0.0
R [120/2] 172.16.0.0
C 192.168.2.0
O [110/110] 192.168.1.0
```

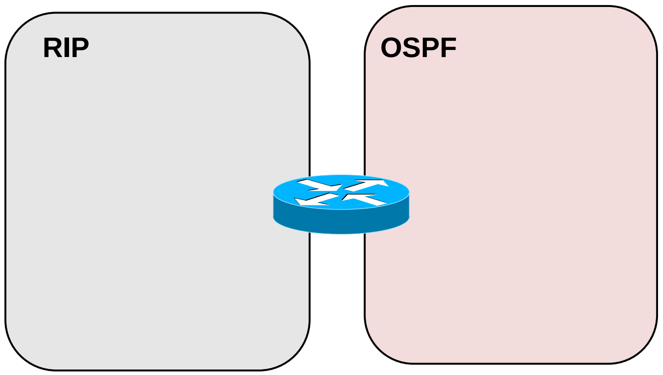
Table R4

```
C 192.168.1.0
C 192.168.2.0
O E2 [110/30] 172.16.0.0
O E2 [110/30] 172.17.0.0
O E2 [110/30] 172.18.0.0
O E2 [110/30] 172.19.0.0
O E2 [110/30] 172.20.0.0
```

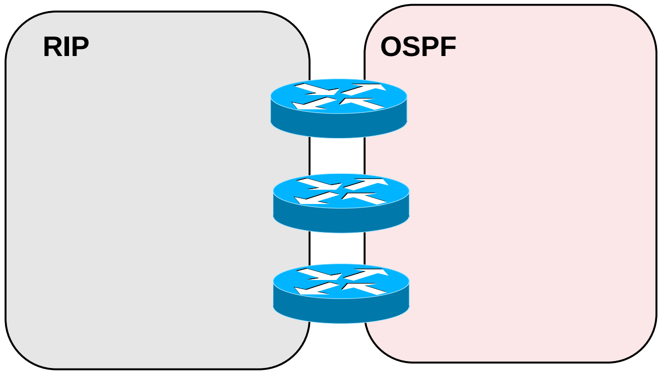
Redistribution Methods

- Redistribution can be done through:
  - **One-point redistribution**
    - Only one router is redistributing one-way or two-way (both ways).
      - There could still be other boundary routers but they are not configured to redistribute.
  - **Multipoint redistribution**
    - Multiple routers are used to redistribute either one-way or two-way (both ways).
    - More prone to routing loop problems.

One-Point Redistribution



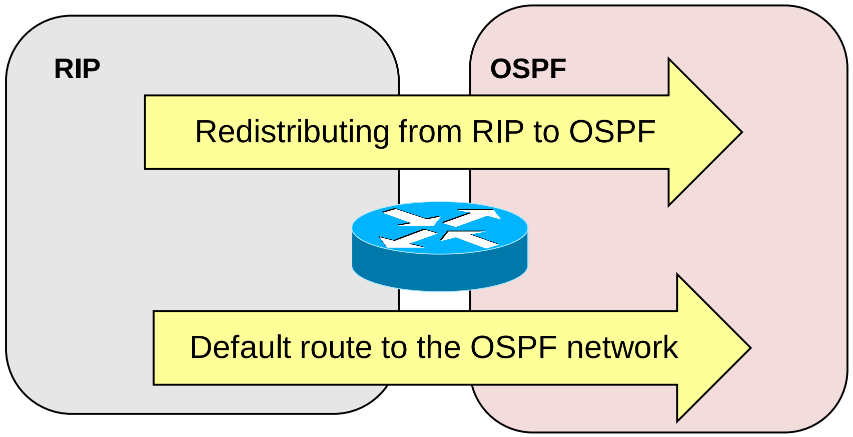
Multipoint Redistribution



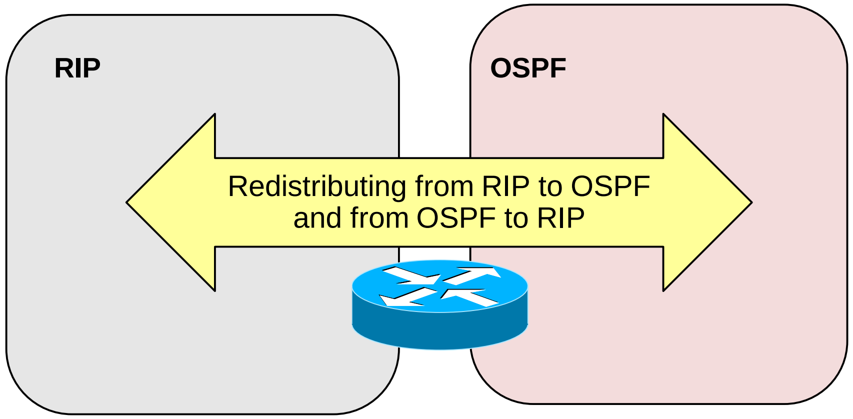
One-Point Redistribution

- One-point redistribution can be configured in either:
  - **One-point One-way**
    - Redistributes networks from one routing protocol into the other routing protocol.
    - Typically uses a default or static route so that devices in that other part of the network can reach the first part of the network.
  - **One-point Two-way**
    - Redistributes routes between the two routing processes, in both directions.

One-Point One-Way Redistribution

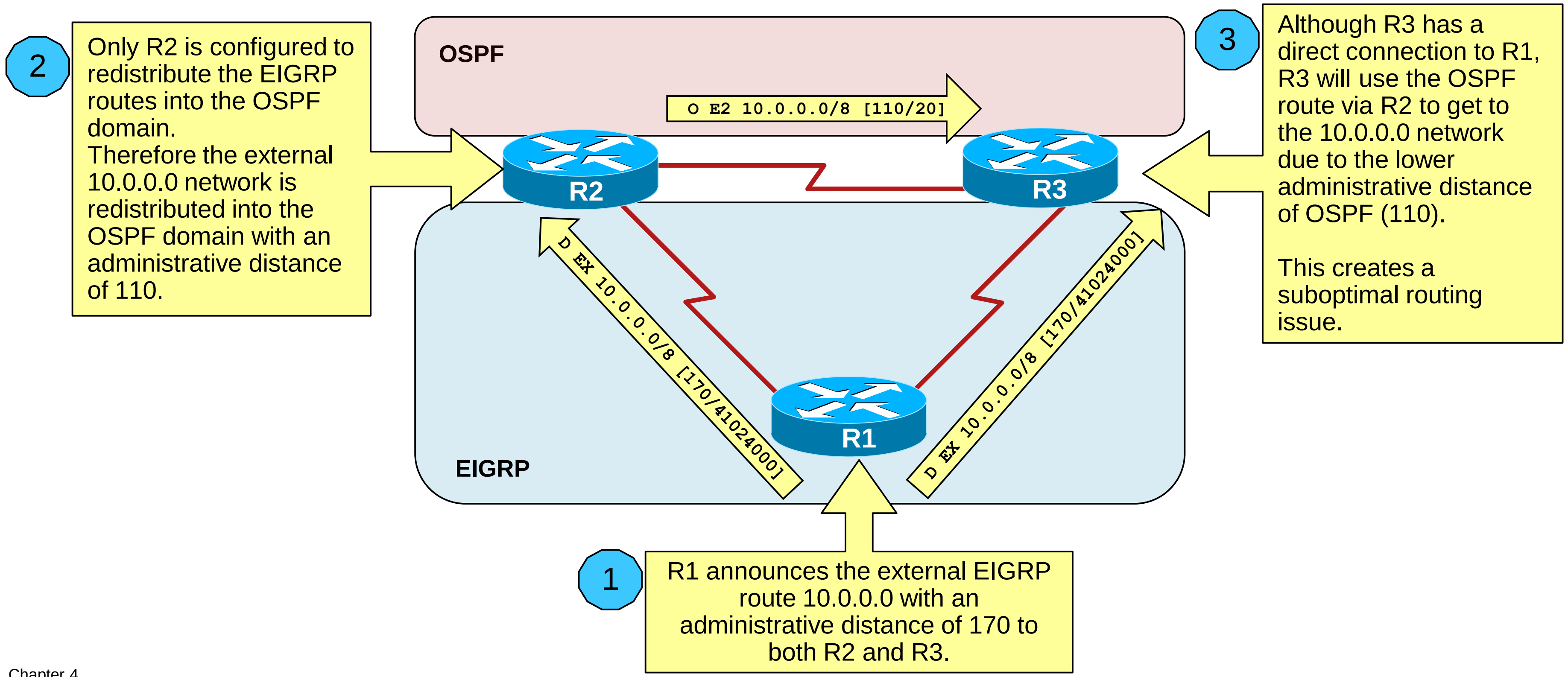

RIP
OSPF
Redistributing from RIP to OSPF
Default route to the OSPF network

One-Point Two-Way Redistribution



One-Point One-Way Redistribution Issue

- Although one-point one-way or two-way redistribution is usually safe from routing loops, issues can still occur if multiple boundary routers exist and only one router is performing one-point one-way redistribution.
  - In this example, R2 is redistributing an external EIGRP route into the OSPF domain.


OSPF
O E2 10.0.0.0/8 [110/20]
R2
R3
R1
D EX 10.0.0.0/8 [170/41024000]
D EX 10.0.0.0/8 [170/41024000]
EIGRP
2
3
1

2 Only R2 is configured to redistribute the EIGRP routes into the OSPF domain. Therefore the external 10.0.0.0 network is redistributed into the OSPF domain with an administrative distance of 110.

3 Although R3 has a direct connection to R1, R3 will use the OSPF route via R2 to get to the 10.0.0.0 network due to the lower administrative distance of OSPF (110).

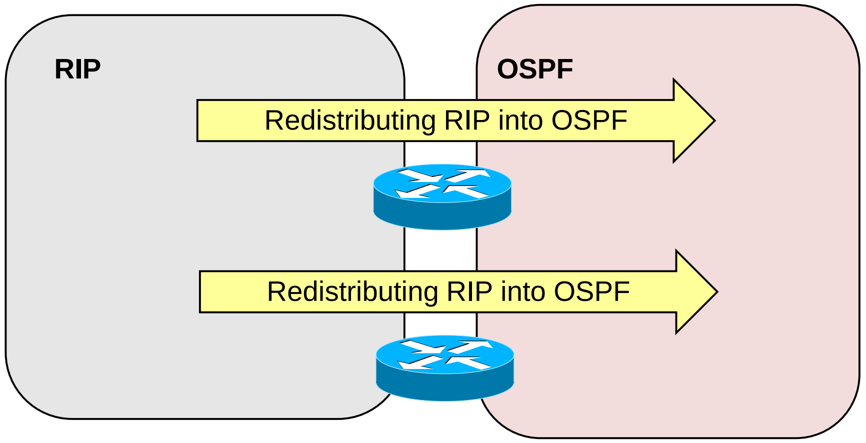
This creates a suboptimal routing issue.

1 R1 announces the external EIGRP route 10.0.0.0 with an administrative distance of 170 to both R2 and R3.

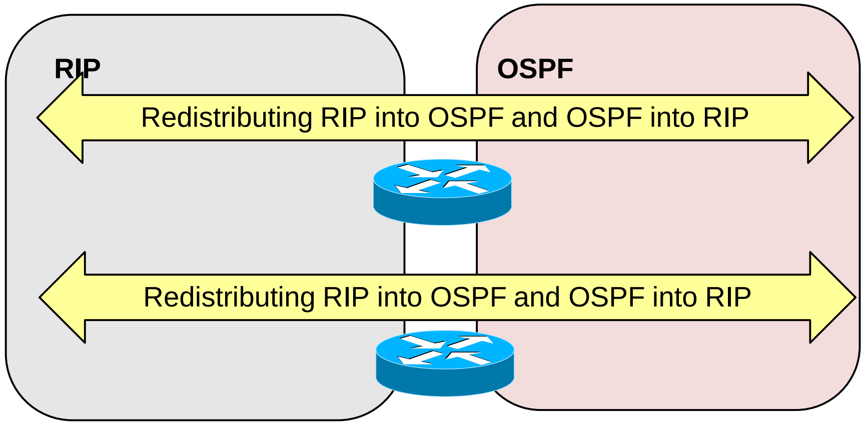
Multipoint Redistribution

- Multipoint redistribution has two (or more) separate routers running both routing protocols.
- Redistribution can be configured as:
  - Multipoint one-way redistribution
  - Multipoint two-way redistribution
- Although multipoint two-way redistribution is especially problematic, either method is likely to introduce potential routing feedback loops.

Multipoint One-Way Redistribution

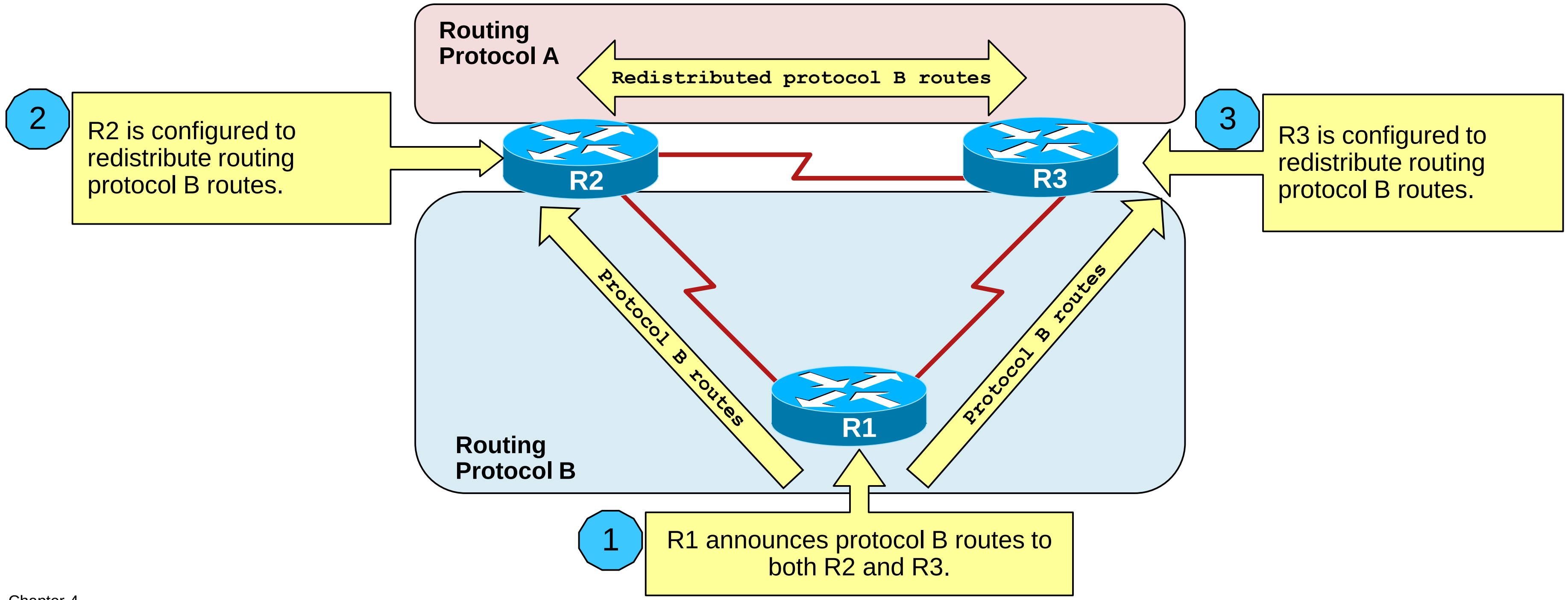


Multipoint Two-Way Redistribution


RIP
OSPF
Redistributing RIP into OSPF and OSPF into RIP
Redistributing RIP into OSPF and OSPF into RIP

Multipoint Redistribution

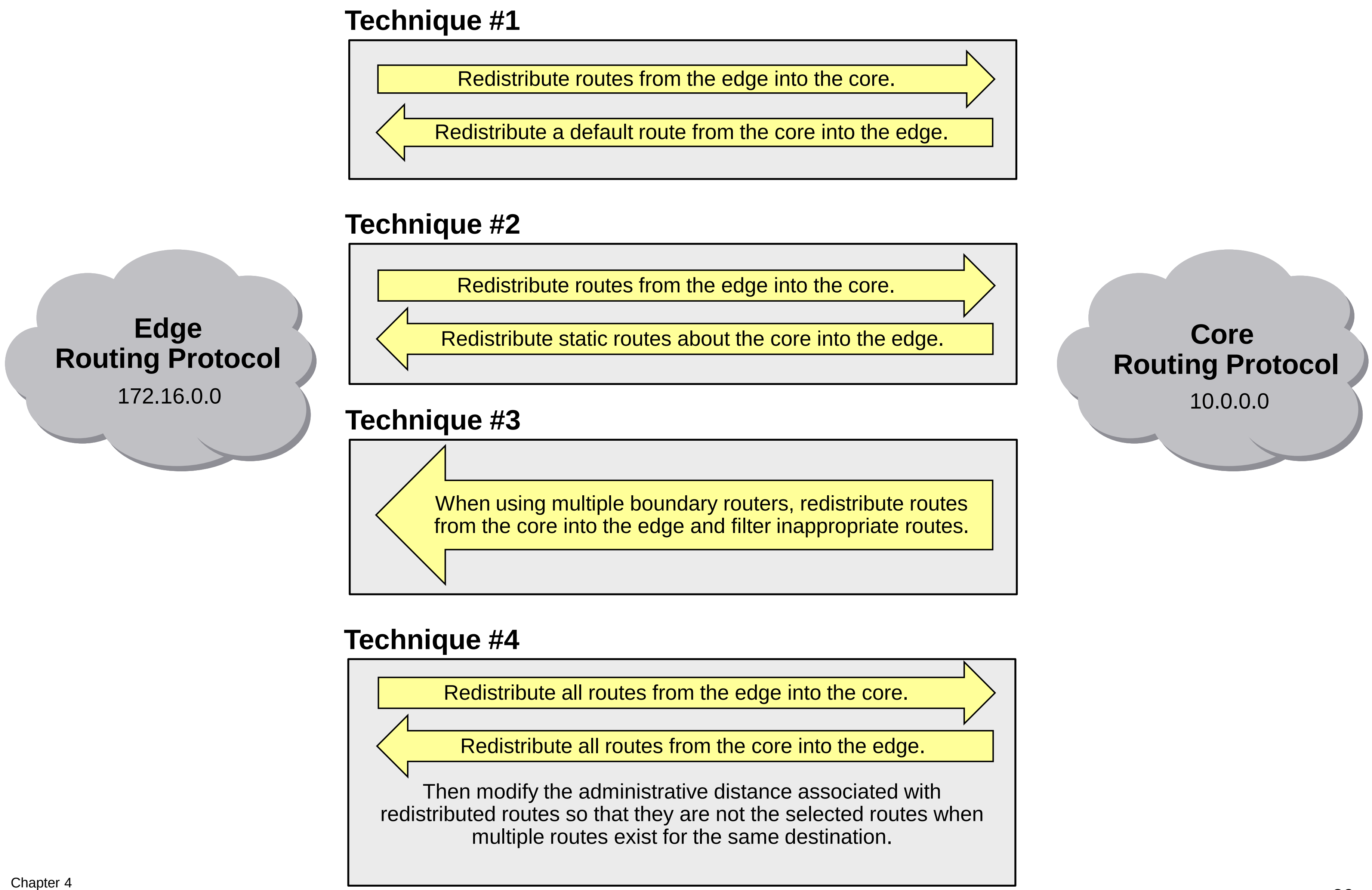
- Multipoint one-way redistribution only works well if:
  - The receiving routing protocol is either EIGRP, BGP and OSPF because they support different administrative distances for internal and external routes.
  - The administrative distance of protocol B's external routes is higher than the administrative distance of protocol A's routes, so that R2 and R3 will use the appropriate routes to destinations in the protocol A side of the network.


Routing Protocol A
Redistributed protocol B routes
2
R2 is configured to redistribute routing protocol B routes.
R2
R3
3
R3 is configured to redistribute routing protocol B routes.
Protocol B routes
Protocol B routes
R1
Routing Protocol B
1
R1 announces protocol B routes to both R2 and R3.

Core and Edge Routing Protocols

- Two terms are often used to distinguish redistribution roles between IGPs:
  - Core routing protocol
  - Edge routing protocol
- In a network that run multiple IGPs:
  - The core routing protocol is the main and more advanced routing protocol running in the network (e.g.; EIGRP, OSPF).
  - The edge routing protocol is the simpler IGP (e.g., RIP).
- If this is an IGP migration from an older IGP to a newer IGP:
  - The core routing protocol is the new routing protocol.
  - The edge routing protocol is the old routing protocol.

Redistribution Techniques



Technique #1

- Redistribute routes from the edge into the core.
- Redistribute a default route from the core into the edge.

Technique #2

- Redistribute routes from the edge into the core.
- Redistribute static routes about the core into the edge.

Technique #3

- When using multiple boundary routers, redistribute routes from the core into the edge and filter inappropriate routes.

Technique #4

- Redistribute all routes from the edge into the core.
- Redistribute all routes from the core into the edge.

Then modify the administrative distance associated with redistributed routes so that they are not the selected routes when multiple routes exist for the same destination.

Preventing Routing Loops

- The safest way to perform redistribution is to redistribute routes in only one direction, on only one boundary router within the network.
  - However, that this results in a single point of failure in the network.
- If redistribution must be done in both directions or on multiple boundary routers, the redistribution should be tuned to avoid problems such as suboptimal routing and routing loops.

Redistribution Guidelines

- Do not overlap routing protocols.
  - Do not run two different protocols in the same Internetwork.
  - Instead, have distinct boundaries between networks that use different routing protocols.
- Be familiar with your network.
  - Knowing the network will result in the best decision being made.

Implementing Route Redistribution

Redistribution Supports All Protocols

```
R1(config)# router rip
R1(config-router)# redistribute ?
  bgp          Border Gateway Protocol (BGP)
  connected    Connected

  isis         ISO IS-IS
  iso-igrp     IGRP for OSI networks
  metric       Metric for redistributed routes
  mobile       Mobile routes
  odr          On Demand stub Routes


  route-map    Route map reference

R1(config-router)# redistribute
```

Key Route Redistribution Points

- Routes are redistributed *into* a routing protocol.
  - Therefore, the **`redistribute`** command is configured under the routing process that is *receiving* the redistributed routes.
- Routes can only be redistributed between routing protocols that support the same protocol stack.
  - For example IPv4 to IPv4 and IPv6 to IPv6.
  - However, IPv4 routes cannot be redistributed into IPv6.
- The method used to configure redistribution varies among combinations of routing protocols.
  - For example, some routing protocols require a metric to be configured during redistribution, but others do not.

Generic Redistribution Steps

1. Identify the boundary router(s) that will perform redistribution.
2. Determine which routing protocol is the core protocol.
3. Determine which routing protocol is the edge protocol.
   - Determine whether all routes from the edge protocol need to be propagated into the core and consider methods that reduce the number of routes.
4. Select a method for injecting the required routes into the core.
   - Summarized routes at network boundaries minimizes the number of new entries in the routing table of the core routers.
5. Consider how to inject the core routing information into the edge protocol.

Redistributing into RIP

- Redistribute routes into RIP.

```
Router(config-router)#
```

```
redistribute protocol [process-id] [match route-type] [metric
  metric-value] [route-map map-tag]
```

Parameter	Description
protocol	The source protocol from which routes are redistributed.
process-id	For OSPF, this value is an OSPF process ID. For EIGRP or BGP, this value is an AS number. This parameter is not required for IS-IS.
route-type	(Optional) A parameter used when redistributing OSPF routes into another routing protocol.
metric-value	(Optional) A parameter used to specify the RIP hop count seed metric for the redistributed route. If this value is not specified and no value is specified using the **`default-metric`** router configuration command, then the default metric is 0 and interpreted as infinity which means that routes will not be redistributed.
map-tag	(Optional) Specifies the identifier of a configured route map to be interrogated to filter the importation of routes from the source routing protocol to the current RIP routing protocol.

Redistributing into RIP Example

```
R1(config)# router rip
R1(config-router)# redistribute ospf 1 metric 3
R1(config-router)#
```

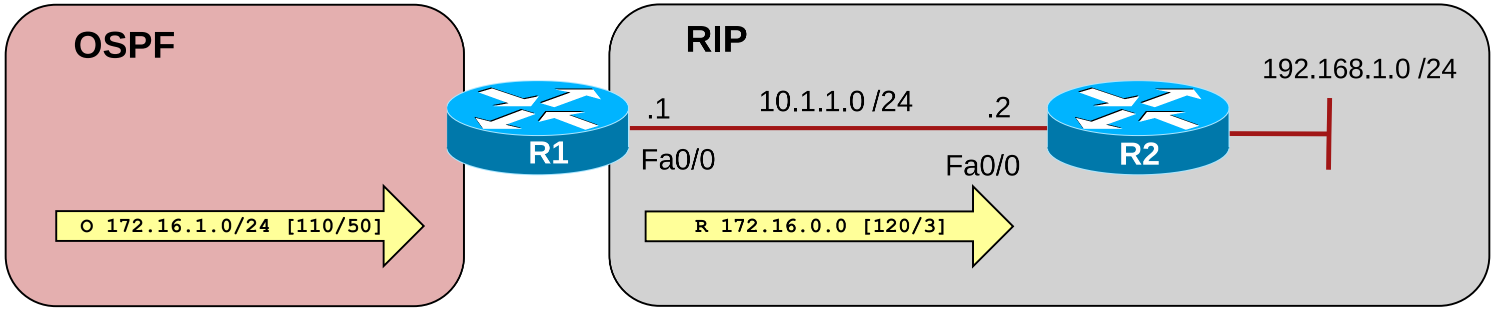


Table R1

```
C 10.1.1.0
R 192.168.1.0 [120/1]
O 172.16.1.0  [110/50]
```

Table R2

```
C 10.1.1.0
C 192.168.1.0
R 172.16.0.0 [120/3]
```

Redistributing into OSPF

- Redistribute routes into OSPF.

```
Router(config-router)#
```

```
redistribute protocol [process-id] [metric metric-value]
  [metric-type type-value] [route-map map-tag] [subnets] [tag
  tag-value]
```

Parameter	Description
protocol	The source protocol from which routes are redistributed.
process-id	For EIGRP or BGP, this value is an AS number. This parameter is not required for RIP or IS-IS.
metric-value	(Optional) A parameter that specifies the OSPF seed metric used for the redistributed route. The default metric is a cost of 20 (except for BGP routes, which have a default metric of 1).
map-tag	(Optional) Specifies the identifier of a configured route map to be interrogated to filter the importation of routes from the source routing protocol to the current OSPF routing protocol.
subnets	(Optional) OSPF parameter that specifies that subnetted routes should be redistributed. Otherwise, only classful routes are redistributed.
tag-value	(Optional) A 32-bit decimal value attached to each external route to be used by ASBRs.

Redistributing into OSPF Example

```
R1(config)# router ospf 1
R1(config-router)# redistribute eigrp 100 subnets metric-type 1
R1(config-router)#
```

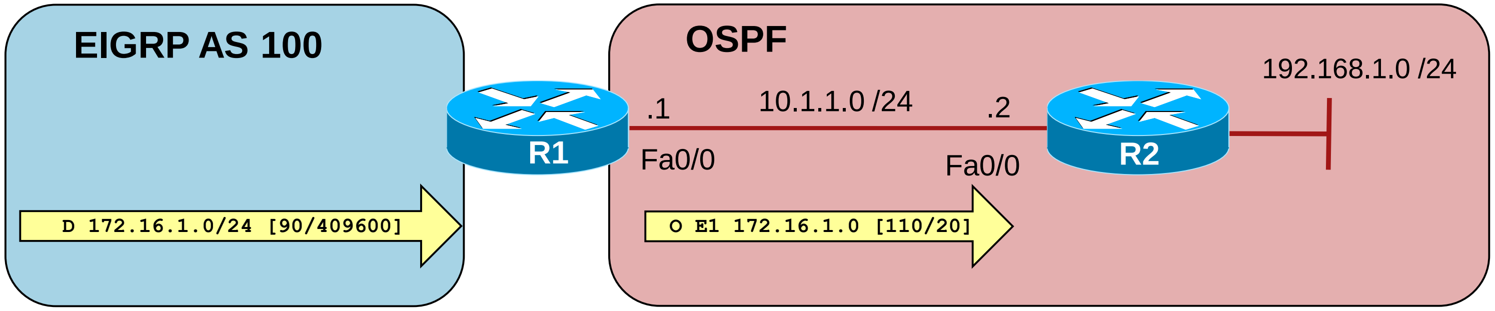


Table R1

```
C 10.1.1.0
O 192.168.1.0  [110/20]
D 172.16.1.0   [90/409600]
```

Table R2

```
C 10.1.1.0
C 192.168.1.0
O E1 172.16.1.0  [110/20]
```

Default Metric for RIP, OSPF, BGP

- Apply default metric values for RIP, OSPF, and BGP.

```
Router(config-router)#
```

```
default-metric number
```

- The *number* parameter is the value of the metric.
  - For RIP this is the number of hops.
  - For OSPF this is the assigned cost.

OSPF Default-Metric Example

```
R1(config)# router ospf 1
R1(config-router)# default-metric 30
R1(config-router)# redistribute eigrp 100 subnets metric-type 1
R1(config-router)#
```

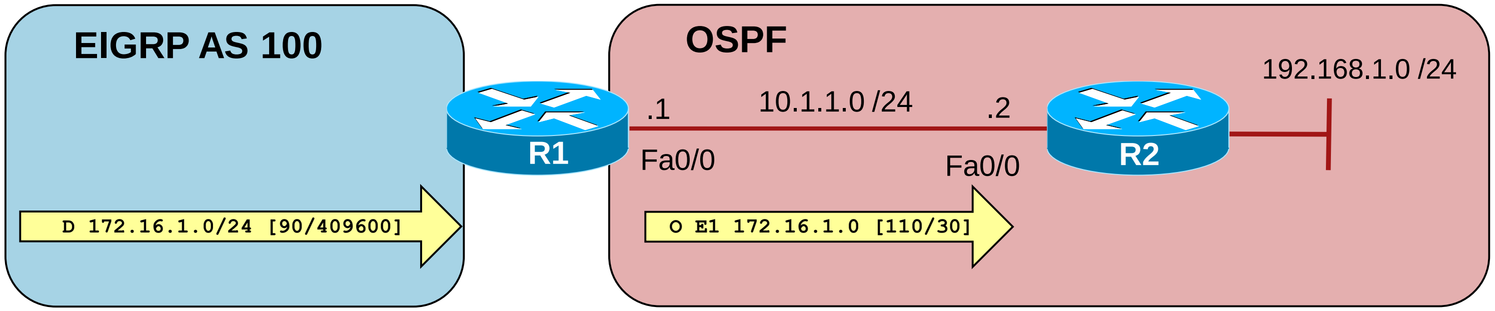


Table R1

```
C 10.1.1.0
O 192.168.1.0 [110/20]
D 172.16.1.0   [90/409600]
```

Table R2

```
C 10.1.1.0
C 192.168.1.0
O E1 172.16.1.0  [110/30]
```

Redistributing into EIGRP

- Redistribute routes into EIGRP.

```
Router(config-router)#
```

```
redistribute protocol [process-id] [match route-type] [metric
  metric-value] [route-map map-tag]
```

Parameter	Description
protocol	The source protocol from which routes are redistributed.
process-id	For OSPF, this value is an OSPF process ID. For BGP, this value is an AS number. This parameter is not required for RIP or IS-IS.
route-type	(Optional) A parameter used when redistributing OSPF routes into another routing protocol.
metric-value	Required if the **default-metric** command is not configured otherwise it is optional . A parameter that specifies the EIGRP seed metric, in the order of bandwidth, delay, reliability, load, and maximum transmission unit (MTU), for the redistributed route. If this value is not specified when redistributing from another protocol and no default metric has been configured, then no routes will not be redistributed.
map-tag	(Optional) Specifies the identifier of a configured route map to be interrogated to filter the importation of routes from the source routing protocol to the current EIGRP routing protocol.

Redistributing into EIGRP Example

```
R1(config)# router eigrp 100
R1(config-router)# redistribute ospf 1 metric 10000 100 255 1 1500
R1(config-router)#
```

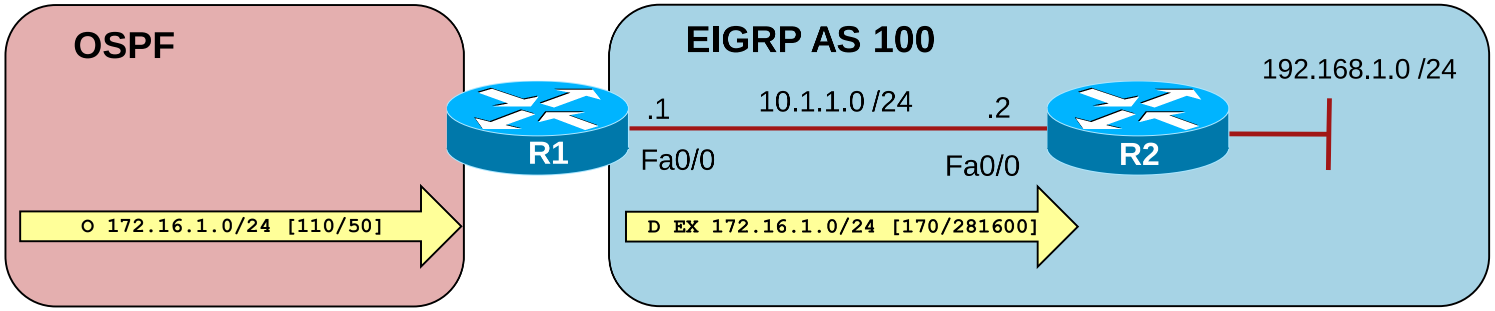


Table R1

```
C 10.1.1.0
O 192.168.1.0 [90/307200]
O 172.16.1.0  [110/50]
```

Table R2

```
C 10.1.1.0
C 192.168.1.0
D EX 172.16.1.0 [170/307200]
```

Default Metric for EIGRP

- Apply metric values for EIGRP.

```
Router(config-router)#
```

```
default-metric bandwidth delay reliability loading mtu
```

Parameter	Description
bandwidth	The route's minimum bandwidth in kilobits per second (kbps). It can be 0 or any positive integer.
delay	Route delay in tens of microseconds. It can be 0 or any positive integer that is a multiple of 39.1 nanoseconds.
reliability	The likelihood of successful packet transmission, expressed as a number from 0 to 255, where 255 means that the route is 100 percent reliable, and 0 means unreliable.
loading	The route's effective loading, expressed as a number from 1 to 255, where 255 means that the route is 100 percent loaded.
mtu	Maximum transmission unit. The maximum packet size in bytes along the route; an integer greater than or equal to 1.

EIGRP Default-Metric Example

```
R1(config)# router eigrp 100
R1(config-router)# default-metric 10000 100 255 1 1500
R1(config-router)# redistribute ospf 1
R1(config-router)#
```

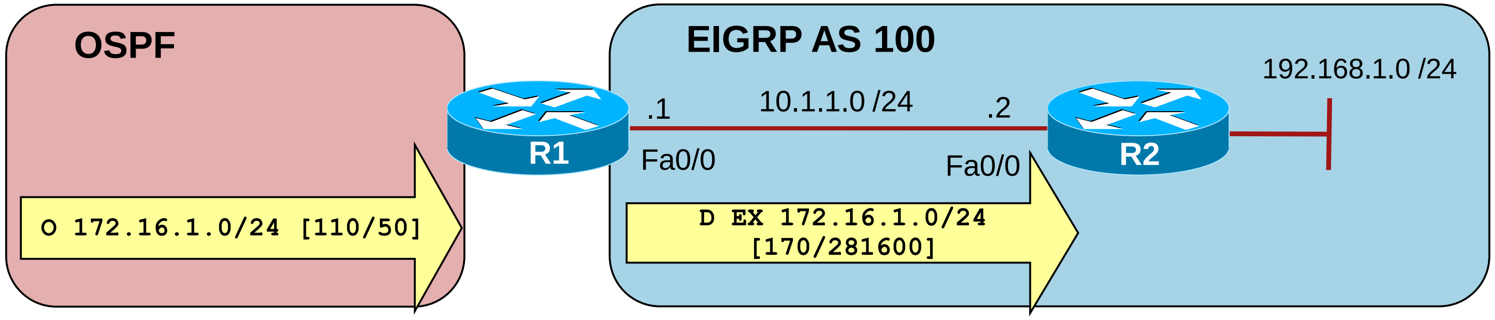


Table R1

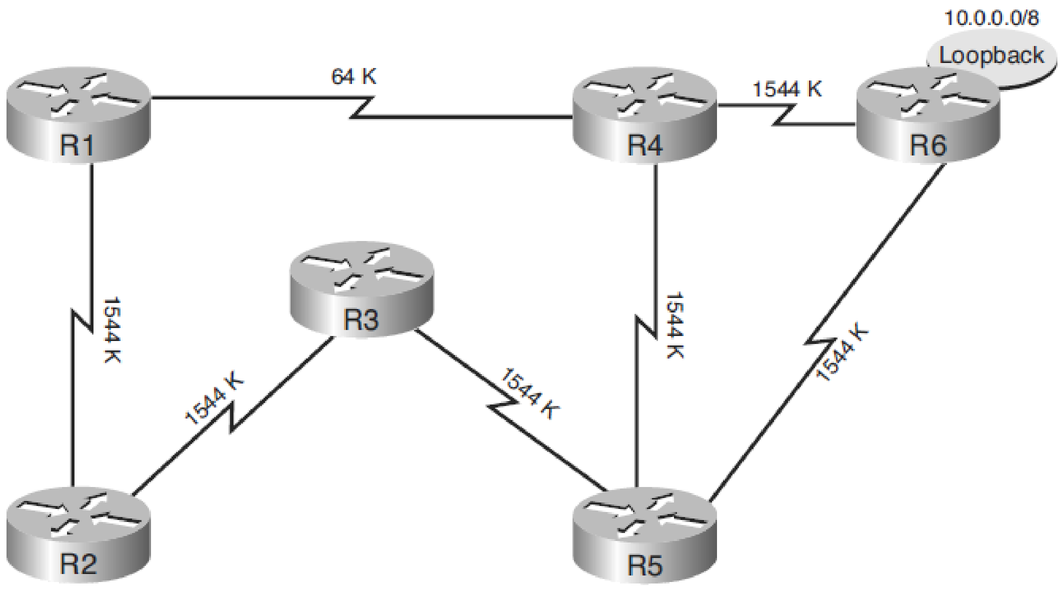
```
C 10.1.1.0
O 192.168.1.0 [90/307200]
O 172.16.1.0  [110/50]
```

Table R2

```
C 10.1.1.0
C 192.168.1.0
D EX 172.16.1.0 [170/307200]
```

Which Path From R1 to 10.0.0.0 /8?

- RIP, OSPF, and EIGRP are all configured on the routers.
- Which path would R1 choose if:
  - RIP made the decision? R1 ⇨ R4 ⇨ R6
  - OSPF made the decision? R1 ⇨ R2 ⇨ R3 ⇨ R5 ⇨ R6
  - EIGRP made the decision? R1 ⇨ R2 ⇨ R3 ⇨ R5 ⇨ R6
- Because EIGRP has the lowest administrative distance of the three protocols, only the EIGRP path to 10.0.0.0/8 is put into the routing table.



Quiz Question

- Assume that a router has three routing processes running simultaneously on it, and each process has received these routes:

EIGRP (internal):	192.168.32.0/26
RIP:	192.168.32.0/24
OSPF:	192.168.32.0/19

- Which of these routes will be installed in the routing table?
- All of them!
  - Although EIGRP has the best administrative distance, each of these routes has a different prefix length (subnet mask).
  - They are therefore considered different destinations and are all installed in the routing table.

Modifying the Administrative Distance

- When routes are redistributed between two different routing protocols, some information may be lost making route selection more confusing.
- One approach to correct this is to control the administrative distance to indicate route selection preference and ensure that route selection is unambiguous.
  - Although, this approach does not always guarantee the best route is selected, only that route selection will be consistent.
- For all protocols use the **`distance`** *`administrative-distance`* router configuration command.
  - Alternatively for OSPF, use the **`distance ospf`** command.
  - Alternatively for EIGRP, use the **`distance eigrp`** command.

Modifying the Administrative Distance

- Change the default administrative distances.

```
Router(config-router)#
```

```
distance administrative-distance [address wildcard-mask [ip-
  standard-list] [ip-extended-list]]
```

Parameter	Description
administrative-distance	Sets the administrative distance, an integer from 10 to 255.
address	(Optional) Specifies the IP address; this allows filtering of networks according to the IP address of the router supplying the routing information.
wildcard-mask	(Optional) Specifies the wildcard mask used to interpret the IP address.
ip-standard-list *ip-extended-list*	(Optional) The number or name of a standard or extended access list to be applied to the incoming routing updates. Allows filtering of the networks being advertised.

Modifying OSPF Administrative Distance

- Change the default administrative distances of OSPF.

```
Router(config-router)#
```

```
distance ospf {[intra-area dist1] [inter-area dist2] [external
  dist3]
```

Parameter	Description
dist1	(Optional) Specifies the administrative distance for all OSPF routes within an area. Acceptable values are from 1 to 255 while the default is 110.
dist2	(Optional) Specifies the administrative distance for all OSPF routes from one area to another area. Acceptable values are from 1 to 255 while the default is 110.
dist3	(Optional) Specifies the administrative distance for all routes from other routing domains, learned by redistribution. Acceptable values are from 1 to 255 while the default is 110.

Modifying EIGRP Administrative Distance

- Change the default administrative distance of EIGRP.

```
Router(config-router)#
```

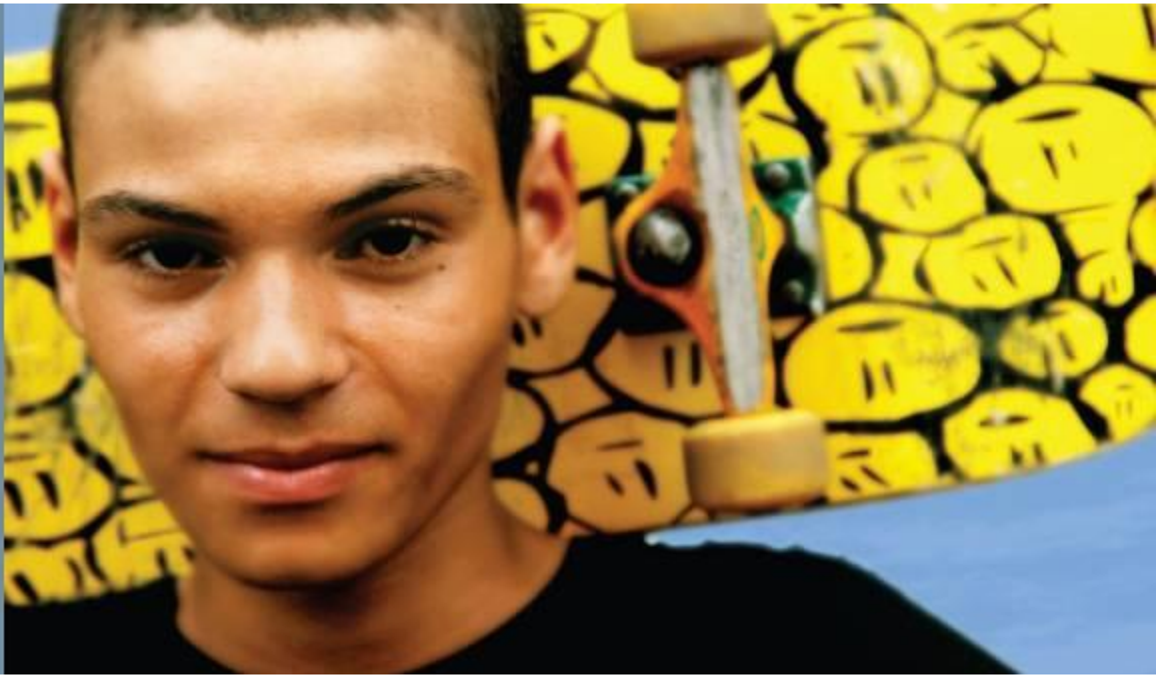
```
distance eigrp internal-distance external-distance
```

Parameter	Description
internal-distance	Specifies the administrative distance for EIGRP internal routes. The distance can be a value from 1 to 255 while the default is 90.
external-distance	Specifies the administrative distance for EIGRP external routes. The distance can be a value from 1 to 255 while the default is 170.

Verifying Redistribution Operation

- Know the network topology.
  - Pay particularly attention to where redundant routes exist.
- Study the routing tables on a variety of routers in the network.
  - For example, check the routing table on the boundary router and on some of the internal routers in each autonomous system.
- Examine the topology table of each configured routing protocol to ensure that all appropriate prefixes are being learned.
- Use the **`traceroute`** EXEC command on some of the routes to verify that the shortest path is being used for routing.
  - Be sure to run traces to networks for which redundant routes exist.
- When troubleshooting, use the **`traceroute`** and **`debug`** commands to observe the routing update traffic on the boundary routers and on the internal routers.

Controlling Routing Update Traffic

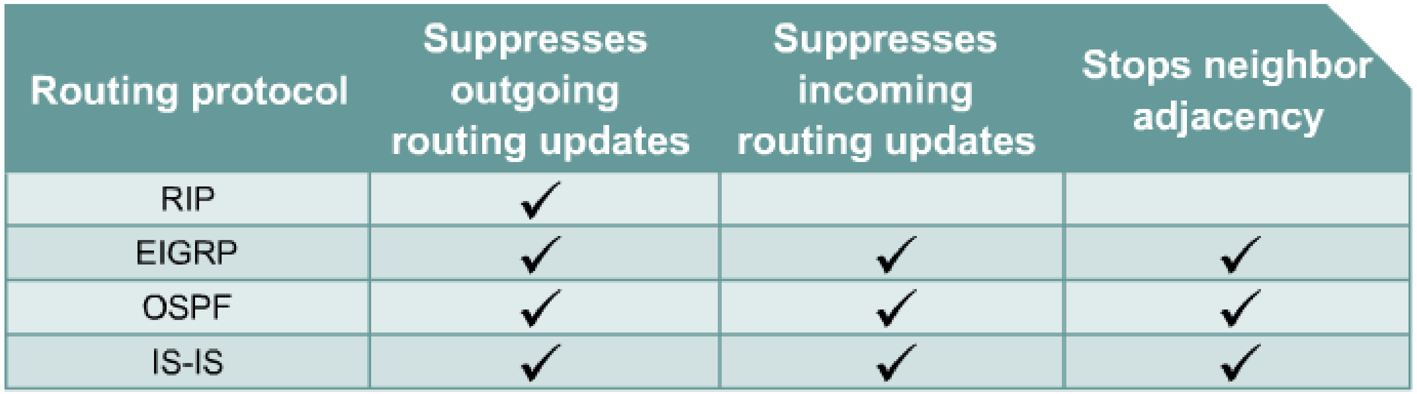


Controlling Routing Updates

- Propagating routing information can be controlled by using:
  - Passive interface
  - Static routes
  - Default route
  - Route maps
  - Distribute lists
  - Prefix lists
- NOTE:
  - There is not one type of route filter that is appropriate for every situation.
  - A variety of techniques may be used to make the network run smoothly.

Passive Interfaces

- Passive interfaces prevent routing updates from being sent and/or received for a specified protocol.
  - RIP interfaces listen but will not send updates.
  - OSPF and EIGRP interfaces do not listen for or send updates and therefore no neighbor adjacencies can be established.



Routing protocol	Suppresses outgoing routing updates	Suppresses incoming routing updates	Stops neighbor adjacency
RIP	✓		
EIGRP	✓	✓	✓
OSPF	✓	✓	✓
IS-IS	✓	✓	✓

passive-interface default Command

- Large enterprise may need to set multiple interfaces as passive.
  - In some networks, this could mean coding 200 or more **`passive-interface`** statements.
- The **`passive-interface default`** command sets all interfaces as passive by default.
  - Interfaces on which adjacencies updates are desired can be set as active with the **`no passive-interface`** command.

Static and Default Routes

- Static routes are manually configured routes that are used to:
  - Define specific routes to use when two autonomous systems must exchange routing information.
  - Define routes to destinations over a WAN link to eliminate the need for a dynamic routing protocol.
- Static route configuration considerations:
  - If you want a router to advertise a static route in a routing protocol, it might need to be redistributed.
  - To reduce the number of static route entries, define a default static route.

Understanding Route Maps

- Route maps are similar in function to ACLs, but provide far more control.
- Route maps are more similar to a scripting language.
  - They can be named rather than numbered for easier documentation.
  - Lines are sequence-numbered for easier editing.
  - Match and set criteria can be used, similar to the “if, then” logic.
    - They allow conditions to be tested using match commands and if the conditions match, actions specified by set commands can be taken to modify attributes of the packet or routes.
- Just as ACLs are used by a variety of Cisco IOS features, route maps can also be used for various applications.
  - The actual route map implementation will vary based on how its applied.

Route Map Applications

- **Route filtering during redistribution**
  - All IP routing protocols can use route maps for redistribution filtering.
  - Applied using the **`redistribute`** *`protocol`* **`route-map`** router configuration command.
- **Policy-based routing (PBR)**
  - PBR allows the operator to define routing policy other than basic destination-based routing using the routing table.
  - Applied using the **`ip policy route-map`** interface configuration command.
- **NAT**
  - Route maps provide more control over which private addresses are translated to public addresses.
- **BGP**
  - Route maps are the primary tools for implementing BGP policy.

Defining a Route Map

- Define a route map and enter route map configuration mode.

```
Router(config)#
```

```
route-map map-tag [permit | deny] [sequence-number]
```

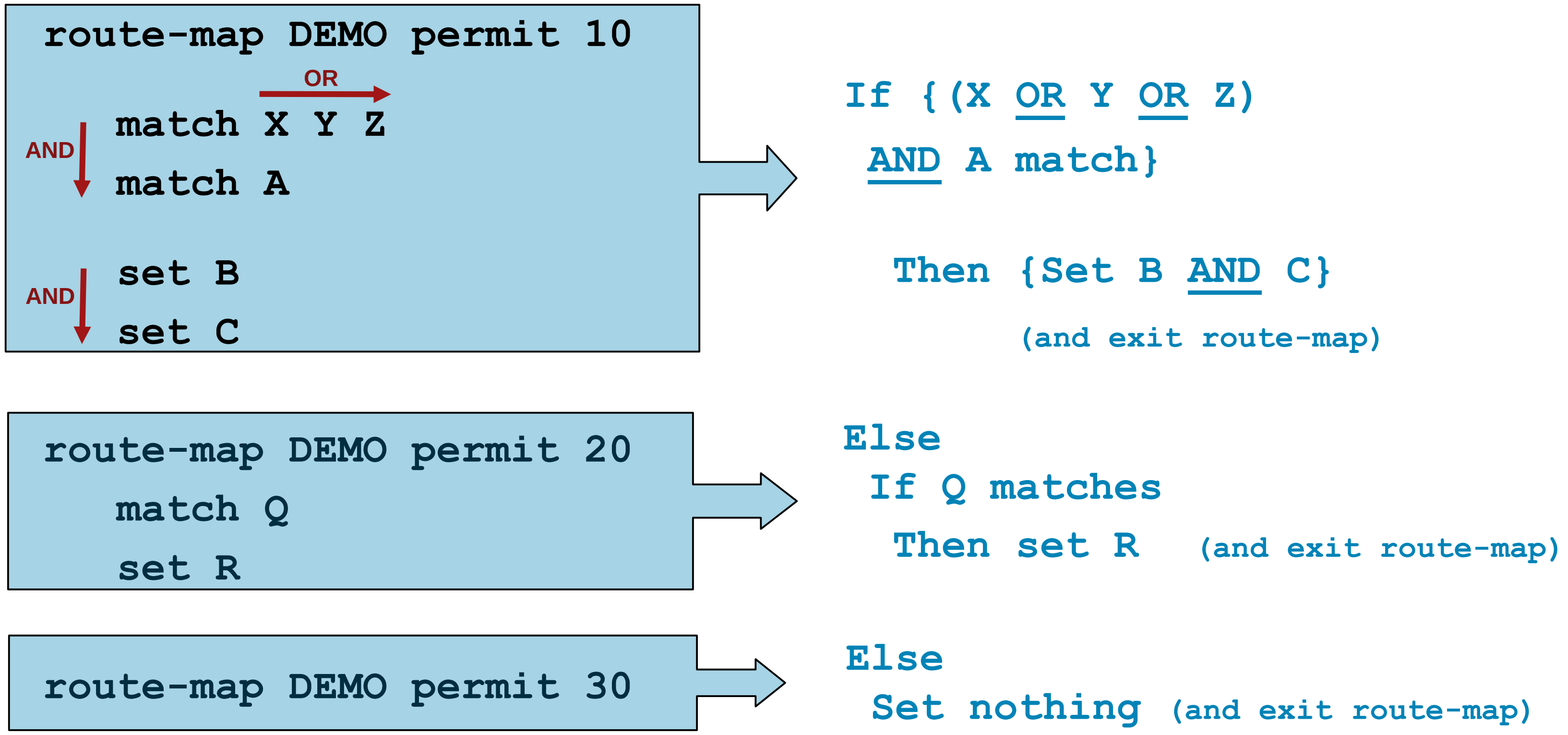
Parameter	Description
map-tag	Name of the route map.
permit | deny	(Optional) A parameter that specifies the action to be taken if the route map match conditions are met; the meaning of permit or deny is dependent on how the route map is used.
sequence-number	(Optional) A sequence number that indicates the position that a new route map statement will have in the list of route map statements already configured with the same name.

- Each **`route map`** statement is numbered by a sequence number and for this reason can be edited.
- The default for the **`route-map`** command is **`permit`**, with a sequence-number of **`10`**.

Route Map Operation Logic

- A route map consists of a list of statements.
  - The list is processed top-down like an access list.
  - Sequence numbers are used for inserting or deleting specific statements.
- Route map permit or deny determines if the candidate will be redistributed.
  - At least one reference must permit the route for it to be a candidate for redistribution.
- The first match found for a route is applied.
  - The match statement may contain multiple references.
    - Multiple match criteria in the same line use a logical OR.
    - Multiple match criteria in multiple separate lines use a logical AND.
  - Once there is a match, set the action (if defined) and leave the route map.
    - Other route-map statements are not processed.

Route Map Operation Example



```
route-map DEMO permit 10
    match X Y Z
    match A

    set B
    set C
```

```
If {(X OR Y OR Z)
 AND A match}

  Then {Set B AND C}
          (and exit route-map)
```

```
route-map DEMO permit 20
    match Q
    set R
```

```
Else
 If Q matches
  Then set R   (and exit route-map)
```

```
route-map DEMO permit 30
```

```
Else
 Set nothing (and exit route-map)
```

- Match criteria on the same line mean a logical OR condition (If this or this or …).
- Multiple match and set criteria on separate lines indicates an AND condition (and if this …).
- A route-map statement without any **`match`** statements will be considered matched.
- Like an access list, an implicit **`deny any`** appears at the end of a route map.
  - The consequences of this deny depend on how the route map is being used.

`match` Statements

- Specify criteria to be matched.

```
Router(config-route-map)#
```

```
match condition
```

- The **`match`** *`condition`* route map configuration commands are used to define the conditions to be checked.
- Some of these conditions are used for BGP policy, some for PBR, and some for redistribution filtering.

The `match` Commands

Command	Description
`match community`	Matches a BGP community
`match interface`	Matches any routes that have the next hop out of one of the interfaces specified
`match ip address`	Matches any routes that have a destination network number address that is permitted by a standard or extended ACL
`match ip next-hop`	Matches any routes that have a next-hop router address that is passed by one of the ACLs specified
`match ip route-source`	Matches routes that have been advertised by routers and access servers at the address that is specified by the ACLs
`match length`	Matches based on the layer 3 length of a packet
`match metric`	Matches routes with the metric specified
`match route-type`	Matches routes of the specified type
`match tag`	Matches tag of a route

`set` Statements

- Modify matching conditions.

```
Router(config-route-map)#
```

```
set action
```

- The command modifies parameters in redistributed routes.
- The specific *`action`* changes or add characteristics, such as metrics, to any routes that have met a **`match`** *`condition`*.

The `set` Commands

Command	Description
`set as-path`	Modifies an AS path for BGP routes
`set automatic-tag`	Computes automatically the tag value
`set community`	Sets the BGP communities attribute
`set default interface`	Indicates where to output packets that pass a match clause of a route map for policy routing and have no explicit route to the destination
`set interface`	Indicates where to output packets that pass a match clause of a route map for policy routing
`set ip default next-hop`	Indicates where to output packets that pass a match clause of a route map for policy routing and for which the Cisco IOS software has no explicit route to a destination
`set ip next-hop`	Indicates where to output packets that pass a match clause of a route map for policy routing
`set level`	Indicates where to import routes for IS-IS and OSPF
`set local-preference`	Specifies a BGP local preference value
`set metric`	Sets the metric value for a routing protocol
`set metric-type`	Sets the metric type for the destination routing protocol
`set tag`	Sets tag value for destination routing protocol
`set weight`	Specifies the BGP weight value

Configuring Route Maps for PBR

- PBR allows the operator to define a routing policy other than basic destination-based routing using the routing table.
  - For example to make packets to take a route other than the obvious shortest path.
- Sample implementation plan:
  - Define and name the route map with the **`route-map`** command.
    - Define the conditions to match (the **`match`** statements).
    - Define the action to be taken when there is a match (the **`set`** statements).
  - Define which interface the route map will be attached to using the **`ip policy route-map`** interface configuration command.
    - PBR is applied to incoming packets.

route-map Commands for PBR

```
Router(config)#
route-map map-tag [permit | deny] [sequence-number]
```

- Defines the route map conditions.

```
Router(config-route-map)#
match {conditions}
```

- Defines the conditions to match.

```
Router(config-route-map)#
set {actions}
```

- Defines the action to be taken on a match.

```
Router(config-if)#
ip policy route-map map-tag
```

- Apply the route-map to the incoming interface.

match Commands Used in PBR

Command	Description
`match community`	Matches a BGP community
`match interface`	Matches any routes that have the next hop out of one of the interfaces specified
`match ip address`	Matches any routes that have a destination network number address that is permitted by a standard or extended ACL
`match ip next-hop`	Matches any routes that have a next-hop router address that is passed by one of the ACLs specified
`match ip route-source`	Matches routes that have been advertised by routers and access servers at the address that is specified by the ACLs
`match length`	Matches based on the layer 3 length of a packet
`match metric`	Matches routes with the metric specified
`match route-type`	Matches routes of the specified type
`match tag`	Matches tag of a route

`set` Commands Used in PBR

Command	Description
`set as-path`	Modifies an AS path for BGP routes
`set automatic-tag`	Computes automatically the tag value
`set community`	Sets the BGP communities attribute
`set default interface`	Indicates where to output packets that pass a match clause of a route map for policy routing and have no explicit route to the destination
`set interface`	Indicates where to output packets that pass a match clause of a route map for policy routing
`set ip default next-hop`	Indicates where to output packets that pass a match clause of a route map for policy routing and for which the Cisco IOS software has no explicit route to a destination
`set ip next-hop`	Indicates where to output packets that pass a match clause of a route map for policy routing
`set level`	Indicates where to import routes for IS-IS and OSPF
`set local-preference`	Specifies a BGP local preference value
`set metric`	Sets the metric value for a routing protocol
`set metric-type`	Sets the metric type for the destination routing protocol
`set tag`	Sets tag value for destination routing protocol
`set weight`	Specifies the BGP weight value

Configuring Route Maps for PBR Example

```
R1(config)# access-list 1 permit 172.21.16.18 0.0.0.0
R1(config)#
R1(config)# route-map MY-ROUTE-MAP permit 10
R1(config-route-map)# match ip address 1
R1(config-route-map)# set ip next-hop 172.30.3.20
R1(config-route-map)#
R1(config-route-map)# interface S0/0/0
R1(config-if)# ip policy route-map MY-ROUTE-MAP
```

- The route map has only one **`permit`** statement.
  - Any packets that match the IP address specified by ACL 1 (172.21.16.18) should be sent to the next hop IP address 172.30.3.20.
- This route map applies to incoming packets on the S0/0/0 interface.

Configuring Route Maps for Redistribution

- Use route maps when you want detailed control over how routes are redistributed between routing protocols.
- Sample implementation plan:
  - Define and name the route map with the **`route-map`** command.
    - Define the conditions to match (the **`match`** statements).
    - Define the action to be taken when there is a match (the **`set`** statements).
  - Specify the route map to use when redistributing.
    - Use the **`redistribute`** *`protocol`* **`route-map`** *`map-tag`* router configuration command.

route-map Commands for Redistribution

```
Router(config)#
route-map map-tag [permit | deny] [sequence-number]
```

- Defines the route map conditions.

```
Router(config-route-map)#
match {conditions}
```

- Defines the conditions to match.

```
Router(config-route-map)#
set {actions}
```

- Defines the action to be taken on a match.

```
Router(config-router)#
redistribute protocol [process-id] route-map map-tag
```

- Allows for detailed control of routes being redistributed into a routing protocol.

`match` Commands Used in Redistribution

Command	Description
`match community`	Matches a BGP community
`match interface`	Matches any routes that have the next hop out of one of the interfaces specified
`match ip address`	Matches any routes that have a destination network number address that is permitted by a standard or extended ACL
`match ip next-hop`	Matches any routes that have a next-hop router address that is passed by one of the ACLs specified
`match ip route-source`	Matches routes that have been advertised by routers and access servers at the address that is specified by the ACLs
`match length`	Matches based on the layer 3 length of a packet
`match metric`	Matches routes with the metric specified
`match route-type`	Matches routes of the specified type
`match tag`	Matches tag of a route

set Commands Used in Redistribution

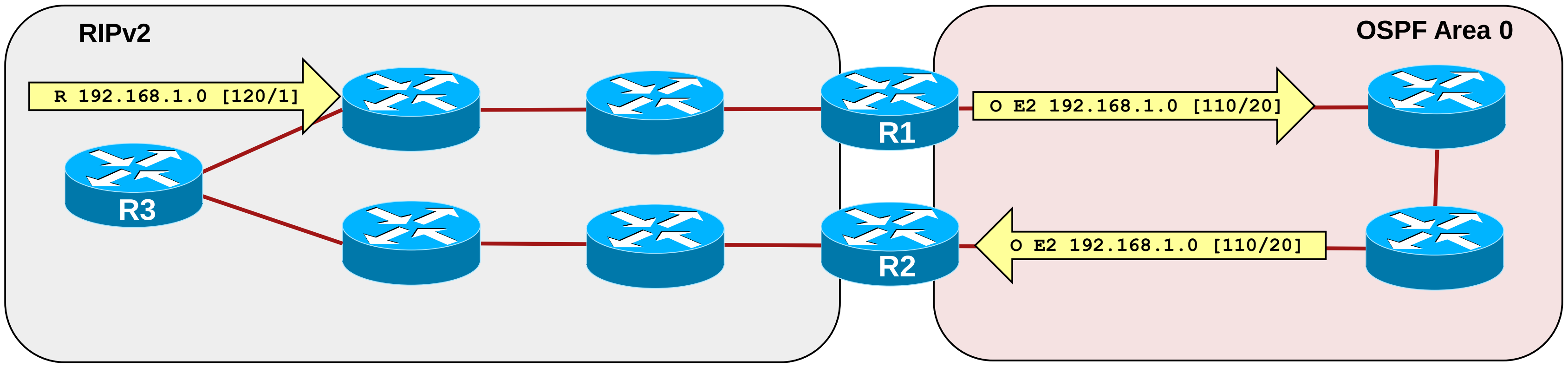
Command	Description
`set as-path`	Modifies an AS path for BGP routes
`set automatic-tag`	Computes automatically the tag value
`set community`	Sets the BGP communities attribute
`set default interface`	Indicates where to output packets that pass a match clause of a route map for policy routing and have no explicit route to the destination
`set interface`	Indicates where to output packets that pass a match clause of a route map for policy routing
`set ip default next-hop`	Indicates where to output packets that pass a match clause of a route map for policy routing and for which the Cisco IOS software has no explicit route to a destination
`set ip next-hop`	Indicates where to output packets that pass a match clause of a route map for policy routing
`set level`	Indicates where to import routes for IS-IS and OSPF
`set local-preference`	Specifies a BGP local preference value
`set metric`	Sets the metric value for a routing protocol
`set metric-type`	Sets the metric type for the destination routing protocol
`set tag`	Sets tag value for destination routing protocol
`set weight`	Specifies the BGP weight value

Configuring Route Maps for Redistribution

```
R1(config)# access-list 23 permit 10.1.0.0 0.0.255.255
R1(config)# access-list 29 permit 172.16.1.0 0.0.0.255
R1(config)# access-list 37 permit 10.0.0.0 0.255.255.255
R1(config)#
R1(config)# route-map REDIS-RIP permit 10
R1(config-route-map)# match ip address 23 29
R1(config-route-map)# set metric 500
R1(config-route-map)# set metric-type type-1
R1(config-route-map)#
R1(config-route-map)# route-map REDIS-RIP deny 20
R1(config-route-map)# match ip address 37
R1(config-route-map)#
R1(config-route-map)# route-map REDIS-RIP permit 30
R1(config-route-map)# set metric 5000
R1(config-route-map)# set metric-type type-2
R1(config-route-map)#
R1(config-route-map)# router ospf 10
R1(config-router)# redistribute rip route-map REDIS-RIP subnets
R1(config-router)#
```

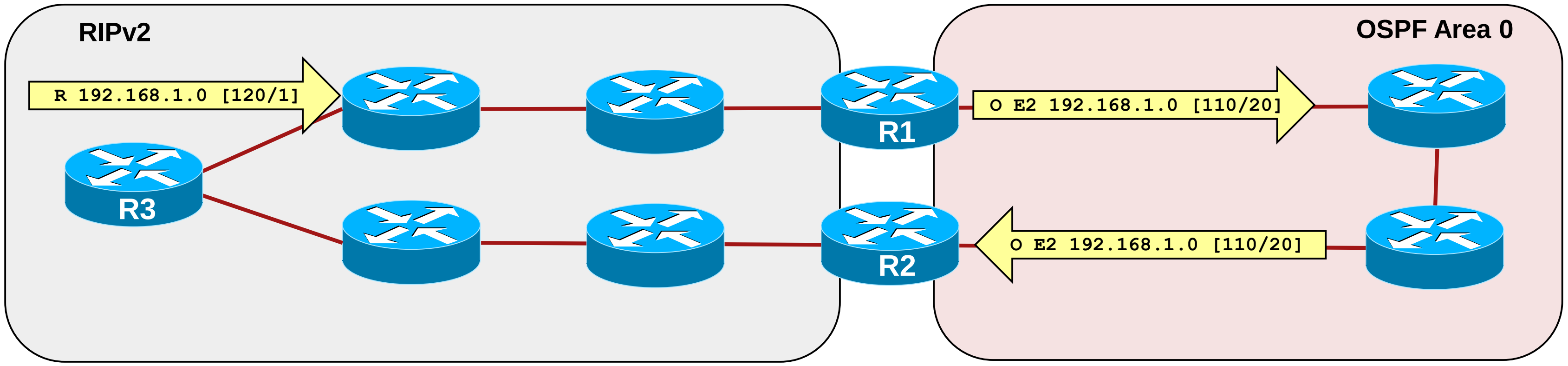
- The route map REDIS-RIP tests the following;
  - In sequence 10, any routes matching ACLs 23 or 29 will have their metric changed accordingly.
  - In sequence 20, any routes matching ACLs 37 will not be redistributed.
  - In sequence 30, all other routes will have their metric changed accordingly.
- Finally, all RIP routes and subnets will be redistributed into OSPF according to the REDIS-RIP route map statements.

Route Feedback



- There is a possibility that routing feedback might cause suboptimal routing when routes are redistributed by more than one router such as in the two-way multipoint redistribution configuration on R1 and R2.
- The following explains the routing feedback loop for this scenario:
  - RIPv2 on R3 advertises network 192.168.1.0.
  - R1 redistributes the 192.168.1.0 network into OSPF.
  - OSPF then propagates this route through the OSPF domain.
  - An OSPF router eventually advertises the 192.168.1.0 network to R2.
  - R2 then redistributes 192.168.1.0 from OSPF back into the original RIPv2 network creating a routing feedback loop.

Route Maps to Avoid Route Feedback



```
R1(config)# access-list 1 permit 192.168.1.0 0.0.0.255
R1(config)# route-map OSPF-into-RIP deny 10
R1(config-route-map)# match ip address 1
R1(config-route-map)# route-map OSPF-into-RIP permit 20
R1(config-route-map)# router rip
R1(config-router)# redistribute ospf 10 metric 5 route-map OSPF-into-RIP
R1(config-router)# router ospf 10
R1(config-router)# redistribute rip subnets
R1(config-router)#
```

- To prevent the routing feedback loop, a route map called OSPF-into-RIP has been applied to R1 and R2.
  - In sequence 10, any routes matching ACL 1 is denied and will not be redistributed back into RIP.
  - In sequence 20, all other routes are permitted to be redistributed and will be assigned a RIP metric of 5.

Using Distribute Lists

- Another way to control routing updates is to use a distribute list which allows an ACL to be applied to routing updates for filtering purposes.
  - Administrators control which routes get distributed.
  - This control is for security, overhead, and management reasons.
- It’s important to understanding that the distribution lists are used to control (filter) routing updates while ACLs filter user traffic.
- Sample implementation plan:
  - Identify network traffic to be filtered using an ACL or route map.
  - Associate the distribute list with the ACL or route-map using the **`distribute-list`** router configuration command.

Filter Incoming Routing Updates

- Define a filter for incoming routing updates.

```
Router(config-router)#
```

```
distribute-list {access-list-number | name} [route-map map-tag] in
   [interface-type interface-number]
```

Parameter	Description
access-list-number | name	Specifies the standard access list number or name.
map-tag	(Optional) Specifies the name of the route map that defines which networks are to be installed in the routing table and which are to be filtered from the routing table. This argument is supported by OSPF only.
in	Applies the access list to incoming routing updates.
interface-type interface-number	(Optional) Specifies the interface type and number from which updates are filtered.

Filter Outgoing Routing Updates

- Define a filter for outgoing routing updates.

```
Router(config-router)#
```

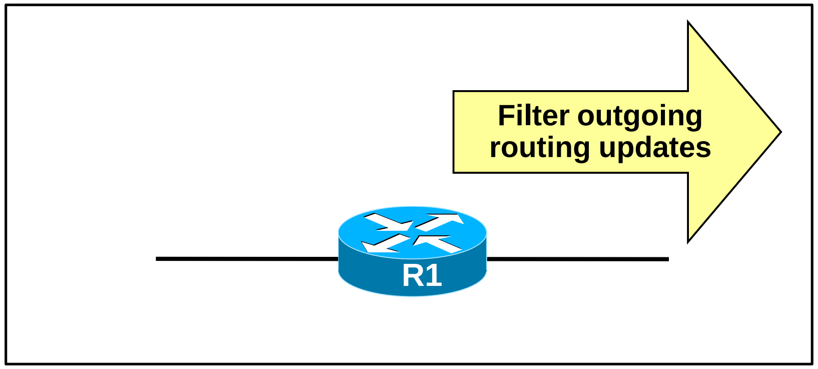
```
distribute-list {access-list-number | name} out [interface-name |
  routing-process [routing-process parameter]]
```

Parameter	Description
access-list-number | name	Specifies the standard access list number or name.
out	Applies the access list to outgoing routing updates.
interface-name	(Optional) Specifies the name of the interface out of which updates are filtered.
routing-process	(Optional) Specifies the name of the routing process, or the keyword **static** or **connected**, that is being redistributed and from which updates are filtered.
routing-process parameter	(Optional) Specifies a routing process parameter, such as the AS number of the routing process.

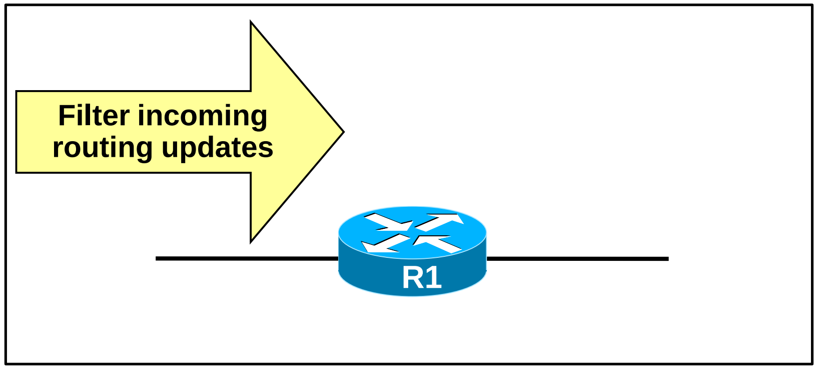
distribute-list out Or in

- It is important to understand the differences between:
  - The **distribute-list out** command filters updates going out of the interface into the routing process under which it is configured.
  - The **distribute-list in** command filters updates going into the interface specified in the command, into the routing process under which it is configured.

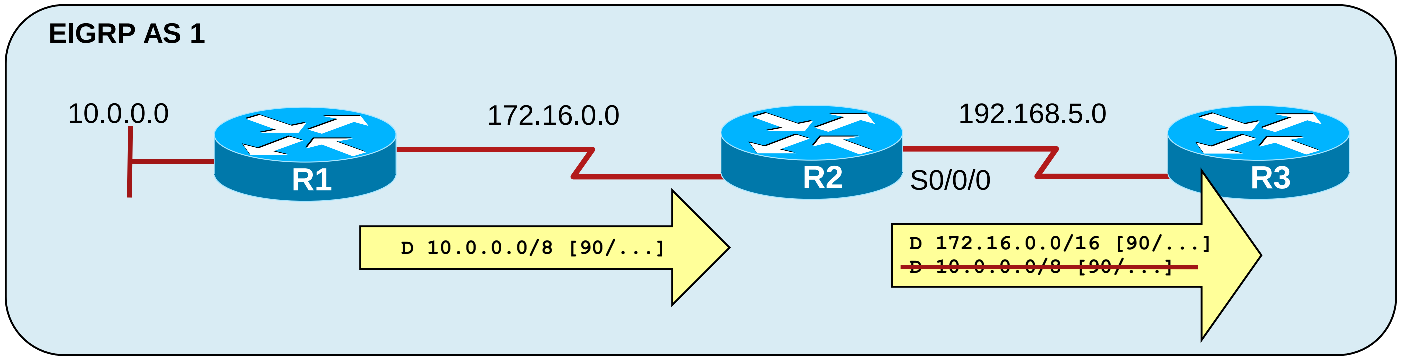
```
R1(config-router)# distribute-list out
```



```
R1(config-router)# distribute-list in
```



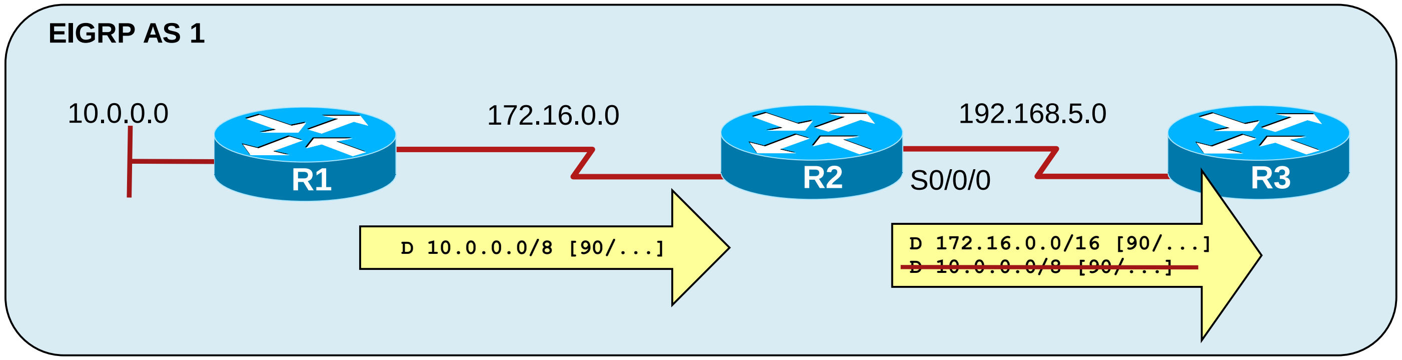
Filter Outgoing Routing Updates Example 1a



```
R2(config)# access-list 7 permit 172.16.0.0 0.0.255.255
R2(config)#
R2(config)# router eigrp 1
R2(config-router)# network 172.16.0.0
R2(config-router)# network 192.168.5.0
R2(config-router)# distribute-list 7 out Serial0/0/0
R2(config-router)#
```

- In this example, the network 10.0.0.0 must be hidden from the devices in network 192.168.5.0.
  - The **distribute-list out** command on R2 applies ACL 7 to packets going out S0/0/0 which only permits 172.16.0.0 routing information to be distributed out.
  - The implicit **deny any** at the end of the ACL prevents updates about any other networks from being advertised and as a result, network 10.0.0.0 is hidden.

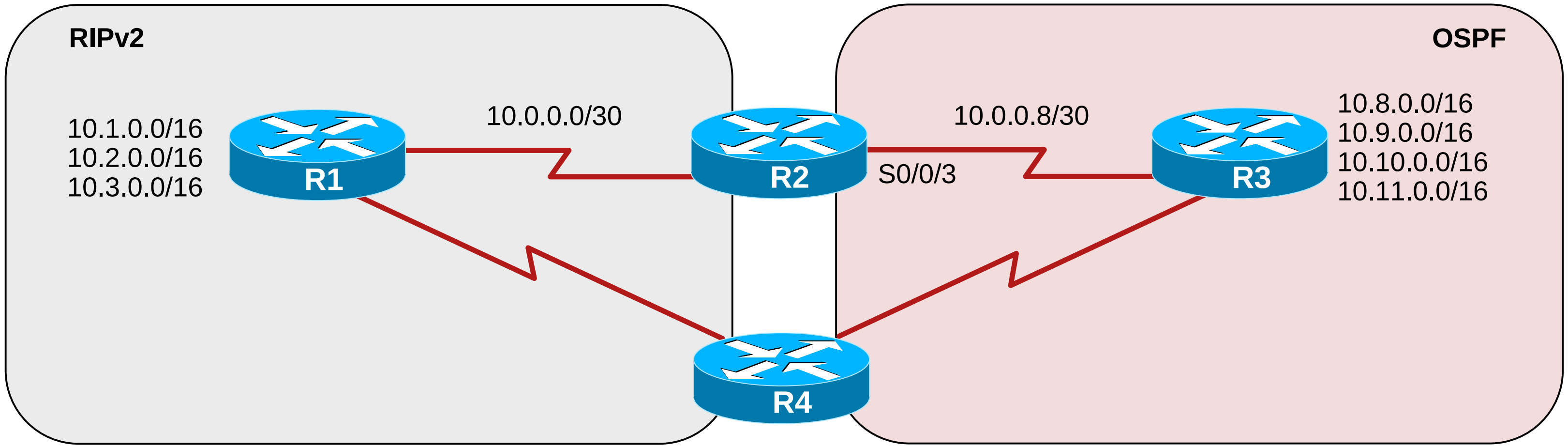
Filter Outgoing Routing Updates Example 1b



```
R2(config)# access-list 7 deny 10.0.0.0 0.255.255.255
R2(config)# access-list 7 permit any
R2(config)#
R2(config)# router eigrp 1
R2(config-router)# network 172.16.0.0
R2(config-router)# network 192.168.5.0
R2(config-router)# distribute-list 7 out Serial0/0/0
R2(config-router)#
```

- As an alternative, network 10.0.0.0 can be explicitly denied and all other routes are valid.
  - The **`distribute-list out`** command on R2 applies ACL 7 to packets going out S0/0/0 which denies the 10.0.0.0/8 network but permits all other routes.

Distribute Lists to Avoid Route Feedback



```
R2(config)# access-list 2 deny 10.8.0.0 0.3.255.255
R2(config)# access-list 2 permit any
R2(config)# access-list 3 permit 10.8.0.0 0.3.255.255
R2(config)# router ospf 1
R2(config-router)# network 10.0.0.8 0.0.0.3 area 0
R2(config-router)# redistribute rip subnets
R2(config-router)# distribute-list 2 out rip
R2(config-router)# router rip
R2(config-router)# network 10.0.0.0
R2(config-router)# version 2
R2(config-router)# passive-interface Serial0/0/3
R2(config-router)# redistribute ospf 1 metric 5
R2(config-router)# distribute-list 3 out ospf 1
R2(config-router)#
```

Drawback of Distribute Lists

- Using distribute lists as route filters has several drawbacks, including:
  - A subnet mask cannot be easily matched.
  - ACLs are evaluated sequentially for every IP prefix in the routing update.
  - An extended ACL can be cumbersome to configure.
  - A distribute list hides network information, which could be considered a drawback in some circumstances.
    - For example, in a network with redundant paths, a distribute list might permit routing updates for only specific paths, to avoid routing loops.
    - In this case, if the primary path goes down, the backup paths are not used because the rest of the network does not know they exist.
    - When redundant paths exist, use other techniques.

Using Prefix Lists

- Prefix lists can be used as an alternative to access lists in many route filtering commands.
- Prefix list characteristics include:
  - A significant performance improvement over ACLs in loading and route lookup of large lists.
  - Support for incremental modifications.
  - An improved user-friendly command-line interface.
  - Greater flexibility in specifying subnet mask ranges.

Similarities Between Prefix Lists and ACLs

- A prefix list can consist of any number of lines, each of which indicates a test and a result.
- When a router evaluates a route against the prefix list, the first line that matches results in either a permit or deny.
- If none of the lines in the list match, the result is “implicitly deny,” just as it is in an access list.

Prefix List Filtering rules

- An empty prefix list permits all prefixes.
- If a prefix is permitted, the route is used. If a prefix is denied, the route is not used.
- Prefix lists consist of statements with sequence numbers. The router begins the search for a match at the top of the prefix list, which is the statement with the lowest sequence number.
- When a match occurs, the router does not need to go through the rest of the prefix list. For efficiency, you might want to put the most common matches (permits or denies) near the top of the list by specifying a lower sequence number.
- An implicit deny is assumed if a given prefix does not match any entries in a prefix list.

Configure a Prefix List

- Define a prefix list.

```
Router(config)#
```

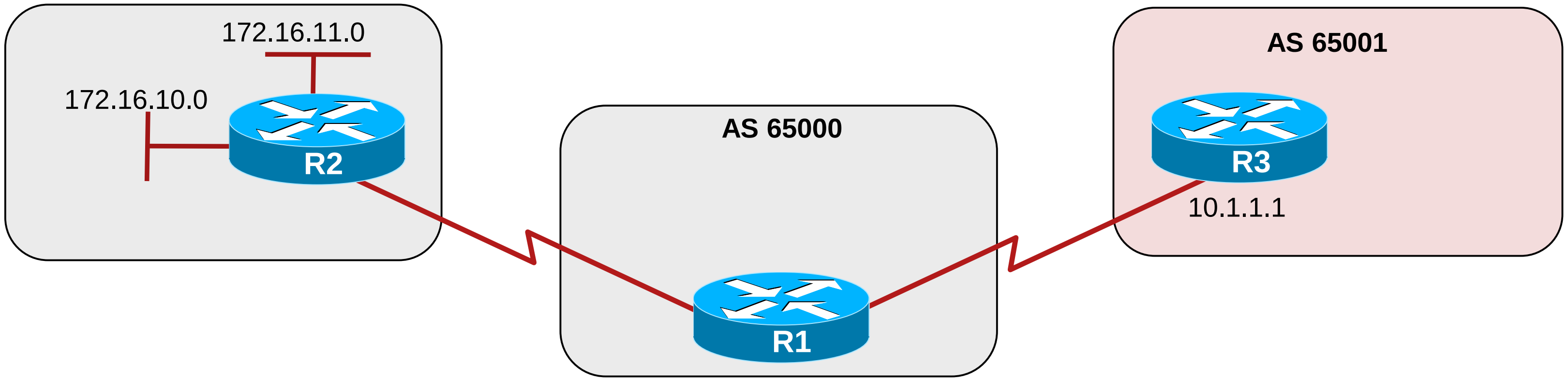
```
ip prefix-list {list-name | list-number} [seq seq-value] {deny |
  permit} network/length [ge ge-value] [le le-value]
```

Parameter	Description
list-name	The name of the prefix list that will be created (it is case sensitive).
list-number	The number of the prefix list that will be created.
seq *seq-value*	A 32-bit sequence number of the **prefix-list** statement. Default sequence numbers are in increments of 5 (5, 10, 15, and so on).
deny | **permit**	The action taken when a match is found.
network / *length*	The prefix to be matched and the length of the prefix. The network is a 32-bit address; the length is a decimal number.
ge *ge-value*	(Optional) The range of the prefix length to be matched. The range is assumed to be from *ge-value* to 32 if only the **ge** attribute is specified.
le *le-value*	(Optional) The range of the prefix length to be matched. The range is assumed to be from length to *le-value* if only the **le** attribute is specified.

Configure a Prefix List

- Use the **`no ip prefix-list`** *`list-name`* global configuration command to delete a prefix list.
- The **`ip prefix-list`** *`list-name`* **`description`** *`text`* global configuration command can be used to add or delete a text description for a prefix list.
- Tip:
  - For best performance, the most frequently processed prefix list statements should be configured with the lowest sequence numbers.
  - The **`seq`** *`seq-value`* keyword can be used for re-sequencing.

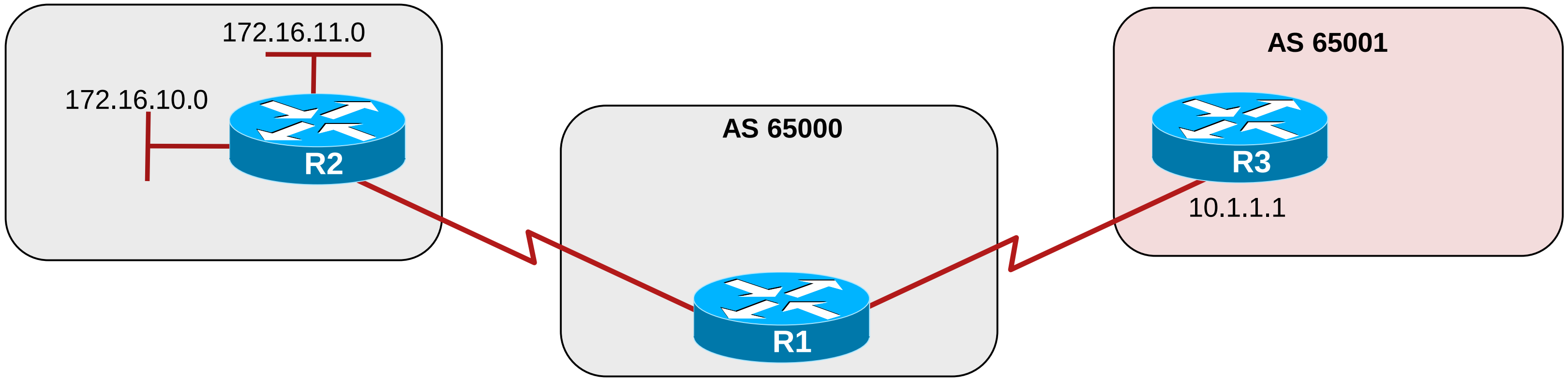
Prefix-list Scenario #1



```
R1(config)# ip prefix-list TEN-ONLY permit 172.16.10.0/8 le 24
R1(config)# router bgp 65000
R1(config-router)# aggregate-address 172.16.0.0 255.255.0.0
R1(config-router)# neighbor 10.1.1.1 remote-as 65001
R1(config-router)# neighbor 10.1.1.1 prefix-list TEN-ONLY out
R1(config-router)# exit
R1(config)# do show running-config | include ip prefix-list
ip prefix-list TEN-ONLY seq 5 permit 172.0.0.0/8 le 24
R1(config)#
```

- Notice that the last line of this configuration changed to **`ip prefix-list TEN-ONLY permit 172.0.0.0/8 le 24`**
  - This is because only the first 8 bits in the address are considered significant when a prefix length of /8 is used.
- In this case, neighbor R3 learns about 172.16.0.0/16, 172.16.10.0/24, and 172.16.11.0/24.
  - These are the routes that match the first 8 bits of 172.0.0.0 and have a prefix length between 8 and 24.

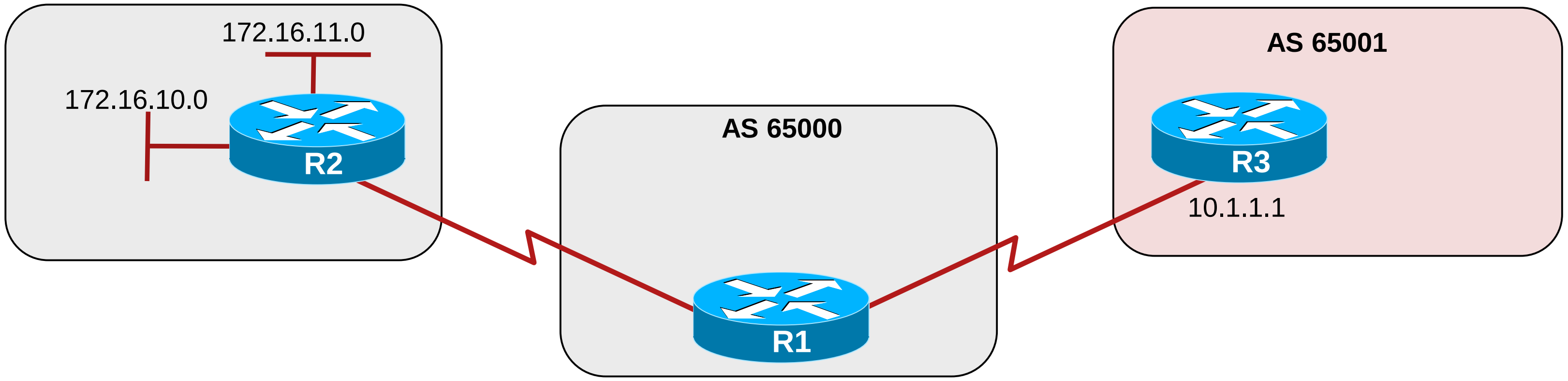
Prefix-list Scenario #2



```
R1(config)# ip prefix-list TEN-ONLY permit 172.16.10.0/8 le 16
R1(config)# router bgp 65000
R1(config-router)# aggregate-address 172.16.0.0 255.255.0.0
R1(config-router)# neighbor 10.1.1.1 remote-as 65001
R1(config-router)# neighbor 10.1.1.1 prefix-list TEN-ONLY out
R1(config-router)# exit
R1(config)#
```

- Now neighbor R3 learns only about 172.16.0.0/16.
  - This is the only route that matches the first 8 bits of 172.0.0.0 and has a prefix length between 8 and 16.

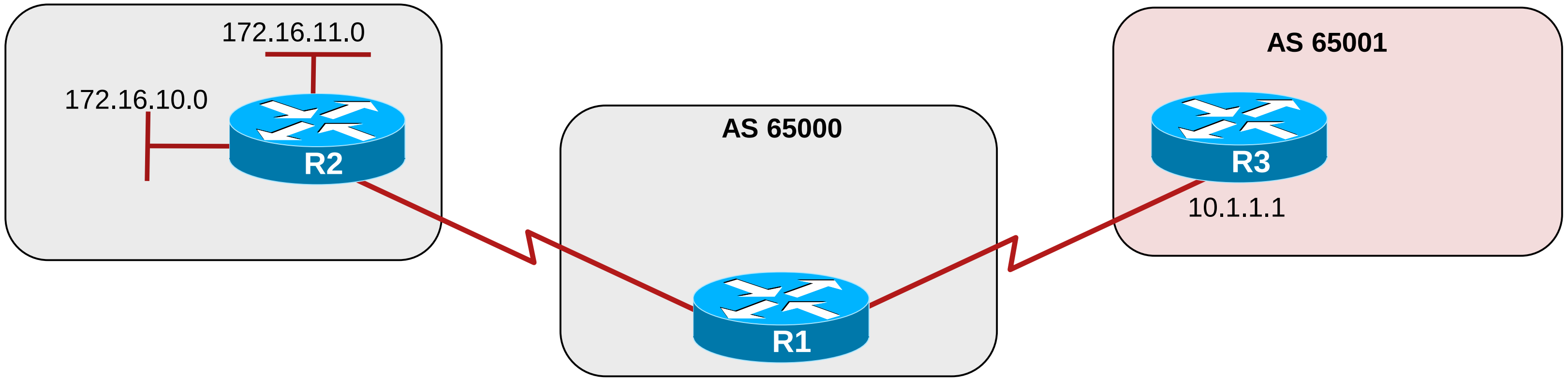
Prefix-list Scenario #3



```
R1(config)# ip prefix-list TEN-ONLY permit 172.16.10.0/8 ge 17
R1(config)# router bgp 65000
R1(config-router)# aggregate-address 172.16.0.0 255.255.0.0
R1(config-router)# neighbor 10.1.1.1 remote-as 65001
R1(config-router)# neighbor 10.1.1.1 prefix-list TEN-ONLY out
R1(config-router)# exit
R1(config)#
```

- Now neighbor R3 learns only about 172.16.10.0/24 and 172.16.11.0/24.
  - R1 ignores the /8 parameter and treats the command as if it had the parameters `ge 17 le 32`.

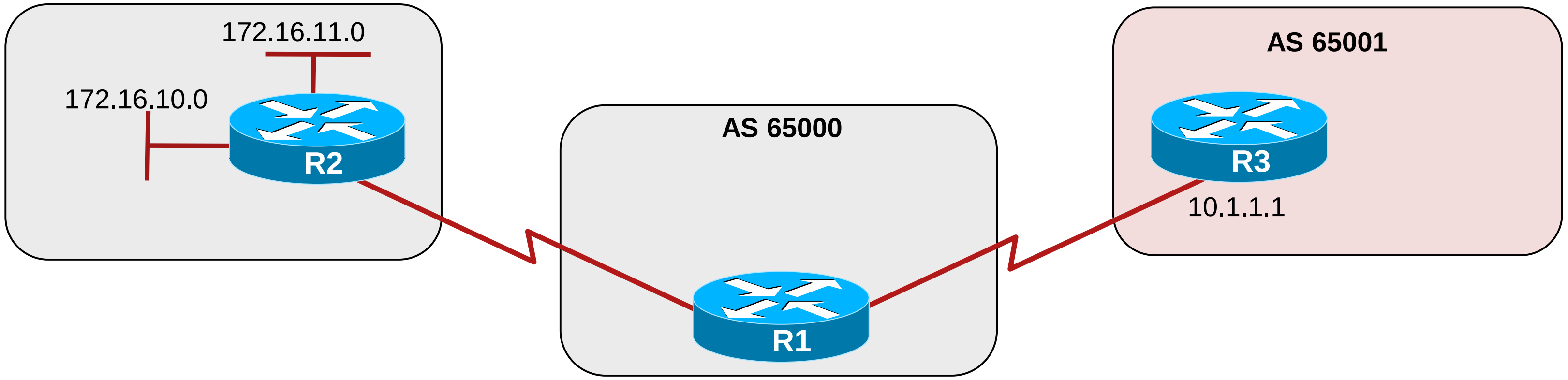
Prefix-list Scenario #4



```
R1(config)# ip prefix-list TEN-ONLY permit 172.16.10.0/8 ge 16 le 24
R1(config)# router bgp 65000
R1(config-router)# aggregate-address 172.16.0.0 255.255.0.0
R1(config-router)# neighbor 10.1.1.1 remote-as 65001
R1(config-router)# neighbor 10.1.1.1 prefix-list TEN-ONLY out
R1(config-router)# exit
R1(config)#
```

- Now neighbor 10.1.1.1 learns about 172.16.0.0/16, 172.16.10.0/24, and 172.16.11.0/24.
  - R1 ignores the /8 parameter and treats the command as if it had the parameters **`ge 16 le 24`**.

Prefix-list Scenario #5



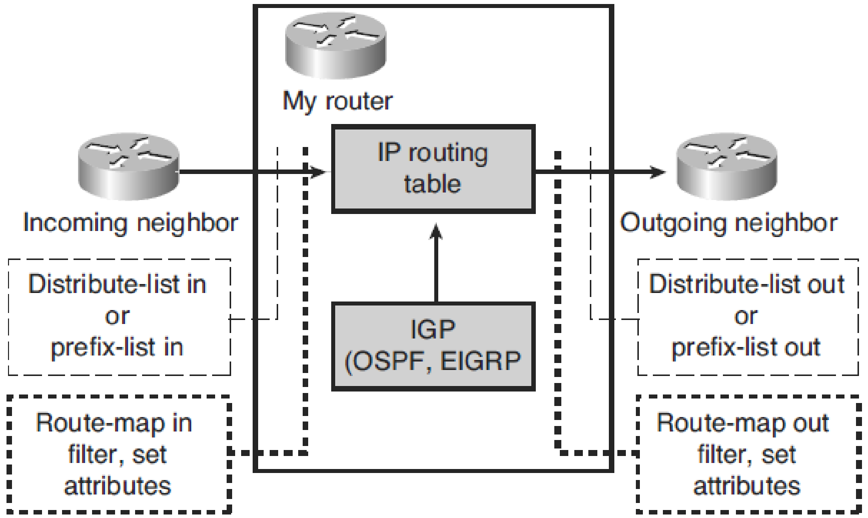
```
R1(config)# ip prefix-list TEN-ONLY permit 172.16.10.0/8 ge 17 le 24
R1(config)# router bgp 65000
R1(config-router)# aggregate-address 172.16.0.0 255.255.0.0
R1(config-router)# neighbor 10.1.1.1 remote-as 65001
R1(config-router)# neighbor 10.1.1.1 prefix-list TEN-ONLY out
R1(config-router)# exit
R1(config)#
```

- Now neighbor 10.1.1.1 learns about 172.16.10.0/24 and 172.16.11.0/24.
  - R1 ignores the /8 parameter and treats the command as if it had the parameters `ge 17 le 24`.

Verifying Prefix Lists

Command	Description
show ip prefix-list [detail | summary]	Displays information on all prefix lists. Specifying the **detail** keyword includes the description and the hit count in the display.
show ip prefix-list [detail | summary] *prefix-list-name*	Displays a table showing the entries in a specific prefix list.
show ip prefix-list *prefix-list-name* **[***network/length***]**	Displays the policy associated with a specific *network/length* in a prefix list.
show ip prefix-list *prefix-list-name* **[seq** *sequence-number***]**	Displays the prefix list entry with a given sequence number.
show ip prefix-list *prefix-list-name* [*network/length*] longer	Displays all entries of a prefix list that are more specific than the given network and length.
show ip prefix-list *prefix-list-name* [*network/length*] first-match	Displays the entry of a prefix list that matches the network and length of the given prefix.
clear ip prefix-list *prefix-list-name* **[***network/length***]**	Resets the hit count shown on prefix list entries.

Multiple Methods to Control Routing Updates


My router
IP routing table
IGP (OSPF, EIGRP
Incoming neighbor
Outgoing neighbor
Distribute-list in or prefix-list in
Route-map in filter, set attributes
Distribute-list out or prefix-list out
Route-map out filter, set attributes

- The example displays how a combination of prefix lists, distribute lists, and route maps can be applied to incoming or outgoing information.
  - All must permit the routes that are received from a neighbor before they will be accepted into the IP routing table.
  - Outgoing routes must pass the outgoing distribute list, the outgoing prefix list, and the outgoing route map before being forwarded to the neighbor.

Chapter 4 Summary

The chapter focused on the following topics:

- Network performance issues and solutions to these issues
  - Includes design changes, passive interfaces, and route filtering (access lists, route maps, distribute lists, and prefix lists).
- Reasons for using more than one routing protocol and how routing information can be redistributed between them.
- How route redistribution is always performed outbound and that the router doing redistribution does not change its routing table.
- Issues arising when redistributing routes, including routing loops, incompatible routing information, and inconsistent convergence times.
- The roles that the administrative distance and the routing metric play in route selection.

Chapter 4 Summary

- When redistributing, a router assigns a seed metric to redistributed routes using the **`default-metric`** router configuration command, or specified as part of the **`redistribute`** command either with the metric option or by using a route map.
- The redistribution techniques, one-point and multipoint.
- Configuration of redistribution between various IP routing protocols.
- Using the **`passive-interface`** router configuration command to prevent routing updates from being sent through the router interface.
- How to manipulate the administrative distance of routes to influence the route selection process.
- Using the **`show ip route`** and **`traceroute`** commands to verify route redistribution.

Chapter 4 Summary

- Using route maps for route filtering during redistribution, PBR, NAT, and BGP.
- The characteristics of route maps and configuration commands including the **`route-map`** *`map-tag`* global configuration command, **`match`** and **`set`** route-map configuration commands.
- Configuring route maps for PBR, using the **`ip policy route-map`** *`map-tag interface`* configuration command.
- Distribute lists, allowing an access list to be applied to routing updates.
- Configuring and verifying prefix lists.

Resources

- Commonly Used IP ACLs

  http://cisco.com/en/US/tech/tk648/tk361/technologies_configuration_example09186a0080100548.shtml

- Default Passive Interface Feature

  http://cisco.com/en/US/products/sw/iosswrel/ps1830/products_feature_guide09186a008008784e.html

- Route-Maps for IP Routing Protocol Redistribution Configuration

  http://cisco.com/en/US/tech/tk365/technologies_tech_note09186a008047915d.shtml

Chapter 4 Labs

- **Lab 4-1 Redistribution Between RIP and OSPF**
- **Lab 4-2 Redistribution Between EIGRP and OSPF**
- **Lab 4-3 Manipulating Administrative Distance**
- **Lab 4-4 EIGRP and OSPF Case Study**

Cisco | Networking Academy®
Mind Wide Open™